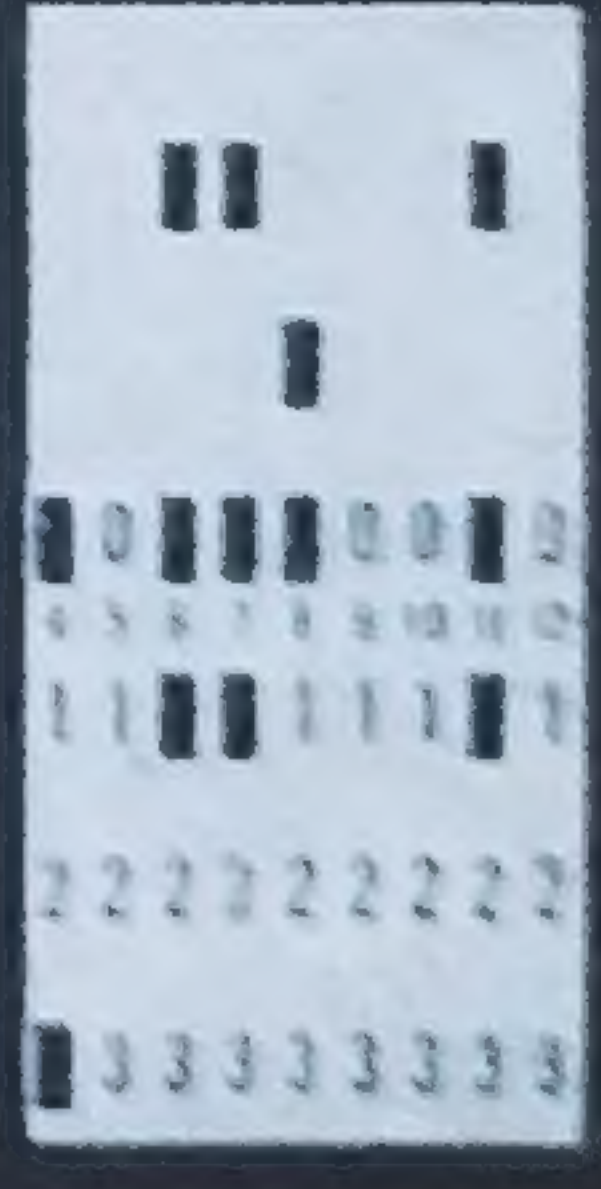
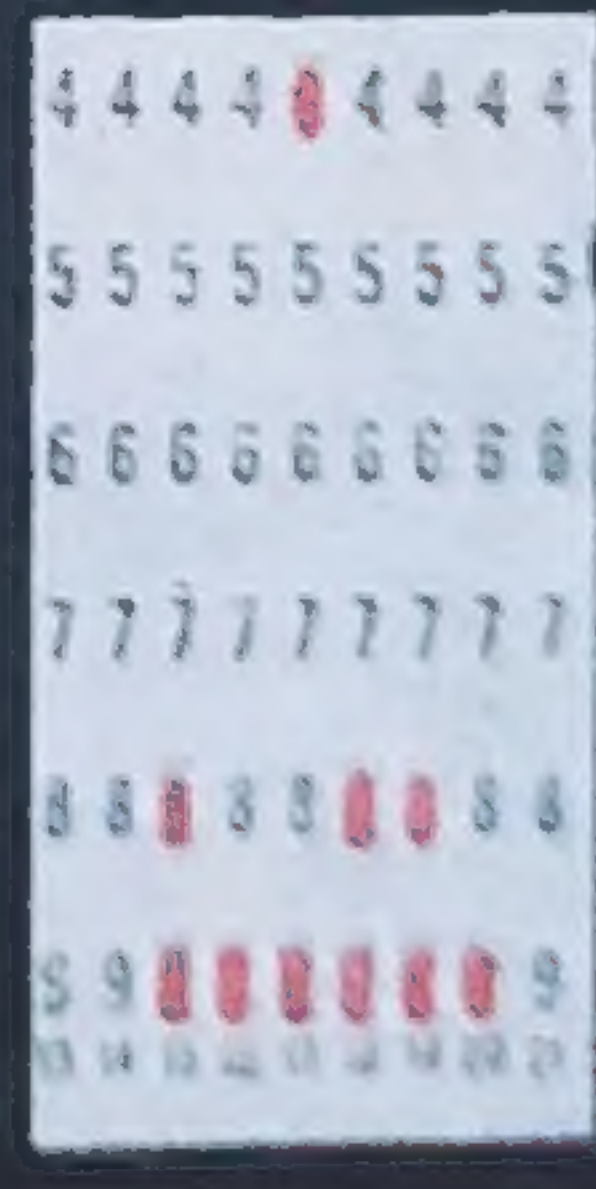
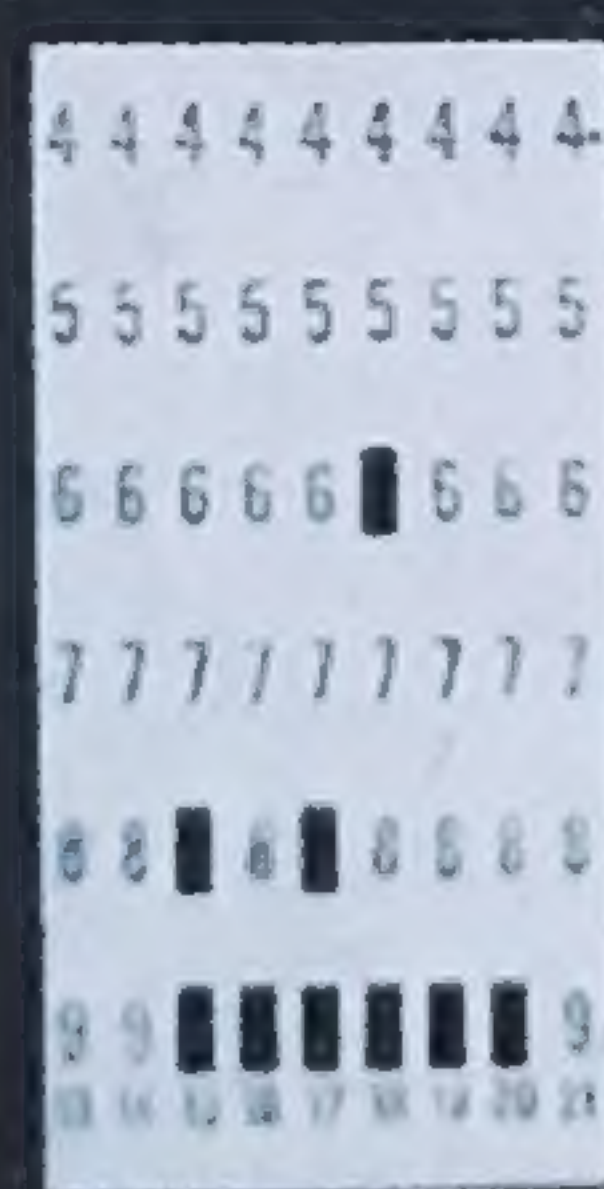
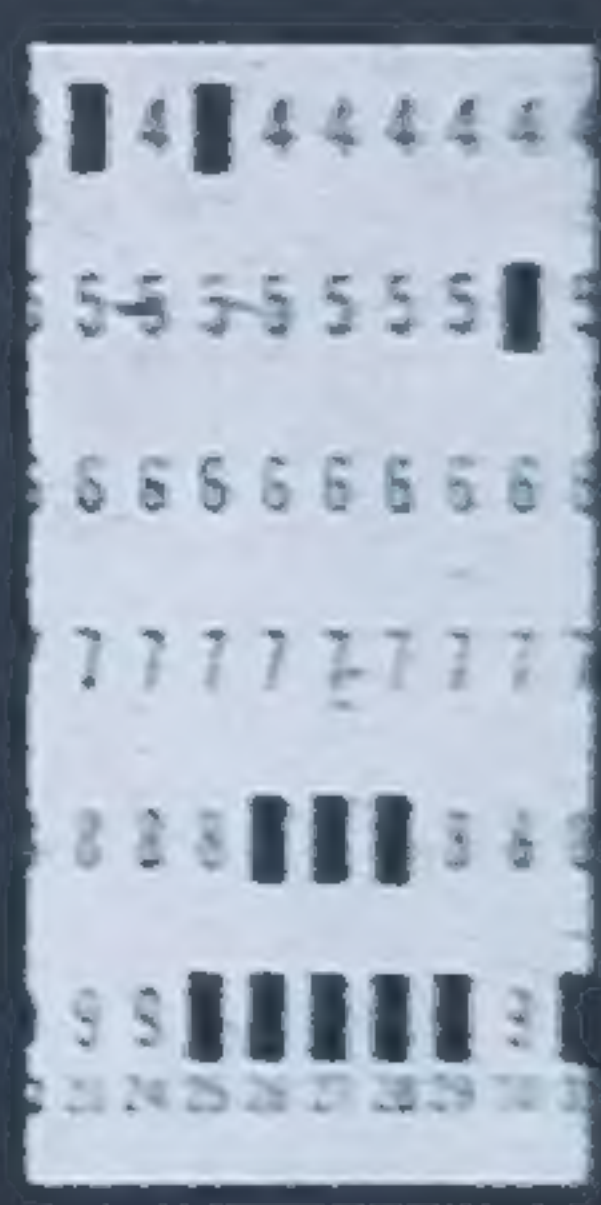
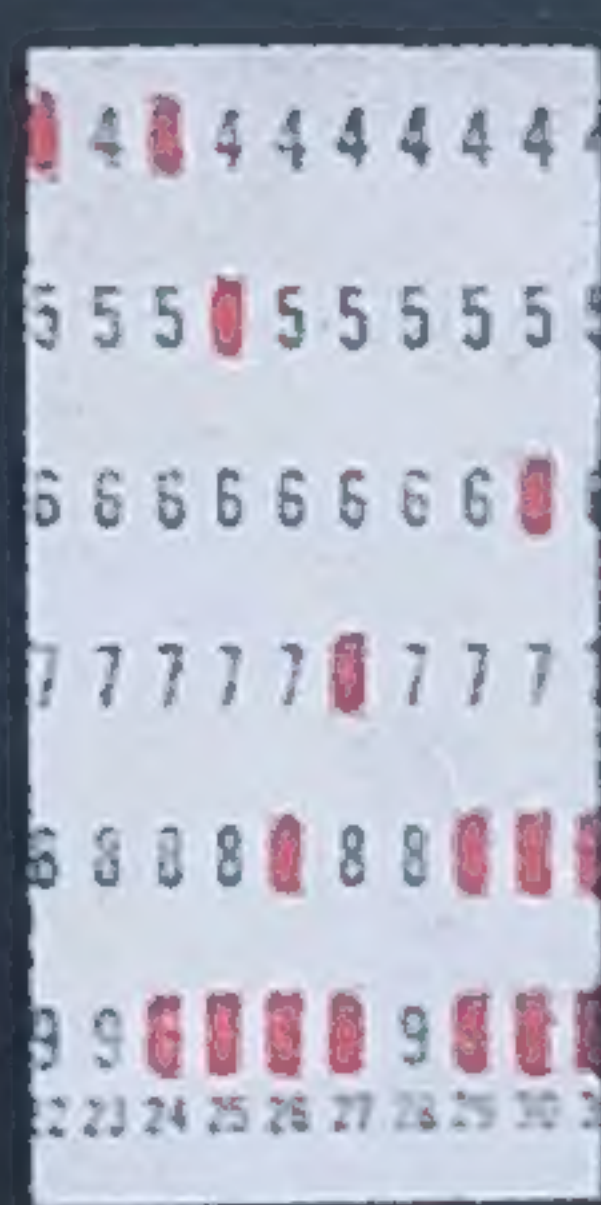


4444444444
5555555555
6666666666
7777777777
8888888888
9999999999
4 5 6 7 8 9 10 11 12
E081 E081



CONSTANTIN VIRGIL NEGOIȚĂ

SISTEME DE ÎNMAGAZINARE
ȘI
REGĂSIRE A INFORMAȚIILOR

EDITURA ACADEMIEI REPUBLICII SOCIALISTE ROMÂNIA
BUCUREȘTI 1970

PREFAȚĂ

Studiul sistemelor de informare, avînd ca obiect analiza, memorarea și regăsirea informațiilor, constituie acum un capitol de bază în domeniul nou și pasionant al aplicațiilor nenumerate ale calculatorului numeric.

Cu cîțiva ani în urmă a devenit limpede că memoria mare și posibilitățile de prelucrare rapidă ale calculatoarelor electronice numerice le recomandă ca singura soluție pentru realizarea unor sisteme de informare, satisfăcătoare, reclamate ca imperios necesare pentru utilizarea rațională a stocurilor mari de informații.

Lucrarea studiază procesul de înmagazinare și regăsire a informațiilor și căile de elaborare a unor modele funcționale care să servească la proiectarea acestor sisteme.

Căutînd să rețină numai rezultatele semnificative într-un domeniu în spectaculoasă evoluție, cartea încearcă să dea o imagine a direcțiilor spre care se focalizează acum efortul de cercetare.

Astfel este prezentată teoria strategiilor de selecție și sînt trecute în revistă unele metode de utilizare a memoriei pentru accelerarea procesului de regăsire.

Fără să presupună cunoștințe de structură a calculatorului sau de metode de programare, cartea se adresează tuturor celor interesați în matematicile aplicate, dar mai ales celor ce se ocupă de aplicațiile calculatorului, adică analiștilor, care au sarcina dificilă de a propune modele pe baza cărora să se proiecteze sisteme folosind structuri disponibile.

Încredințat că lucrarea va fi utilă celor ce au de rezolvat probleme de regăsire a informațiilor, mulțumesc pe această cale Academiei Republicii Socialiste România, care a sprijinit apariția acestei cărți.

C. V. NEGOIȚĂ

27 octombrie 1969

CUPRINS

Introducere	11
CAPITOLUL 1. Înmagazinarea documentelor	19
1.1. Procesul de indexare	19
1.2. Criterii de indexare automată	20
1.2.1. Criteriul statistic	20
1.2.2. Criteriul sintactic	21
1.3. Înregistrarea documentelor	23
1.3.1. Vectorul înregistrare.	23
1.3.2. Matricea de fixare	24
1.4. Observații bibliografice	24
CAPITOLUL 2. Strategii de selecție	27
2.1. Funcția de selecție	27
2.2. Funcția de selecție ca restricție a măsurii exterioare	28
2.3. Funcția de selecție ca funcție etajată	29
2.4. Construcția funcției de selecție	29
2.5. Strategia de selecție	31
2.5.1. Strategii cu funcții aditive	32
2.5.2. Strategii cu funcții de apropiere	33
2.5.3. Strategii cu funcții de repartiție	37
2.6. Echivalența strategiilor de selecție	40
2.6.1. Deosebirea de răspuns	40
2.6.2. Lungimea de selecție	41
2.7. Eficacitatea strategiilor de selecție	42
2.8. Răspunsul sistemului	45
2.9. Strategii cu cerere structurată	47
2.10. Observații bibliografice	49
CAPITOLUL 3. Sisteme cu selecție prin asociere	51
3.1. Matricea de similitudine	51

3.2.	Metoda extinderii cererii de selecție	52
3.3.	Metoda valorii de asociere	53
3.4.	Factori de similitudine	54
3.4.1.	Factori de similitudine de tip δ	54
3.4.2.	Factori de similitudine de tip α	57
3.5.	Factori de sinonimie	58
3.5.1.	Factori de sinonimie bazați pe matricea de similitudine	58
3.5.2.	Factori de sinonimie bazați pe context	59
3.6.	Reducerea vocabularului	60
3.7.	Observații bibliografice	66
CAPITOLUL 4.	Organizarea colecției	67
4.1.	Sisteme cu căutare secvențială	68
4.2.	Metode de scurtare a timpului de interogare	69
4.2.1.	Organizarea în listă	74
4.2.2.	Organizarea în dicționar	79
4.2.3.	Memorii asociative	88
4.3.	Observații bibliografice	95
CAPITOLUL 5.	Sisteme cu clasificare automată	97
5.1.	Matricea de similitudine între înregistrări	97
5.1.1.	Matricea S	97
5.1.2.	Matricea K	99
5.2.	Metoda predispozițiilor	100
5.3.	Metoda grafelor neorientate	106
5.4.	Metoda distanței minime	113
5.4.1.	Funcții discriminante liniare	113
5.4.2.	Clasificarea cu matrice instruibile	115
5.4.3.	Caracterul invariant al clasificării	119
5.4.4.	Proprietatea corecției automate	120
5.5.	Metoda funcțiilor de apartenență	121
5.5.1.	Funcția de apartenență	121
5.5.2.	Utilizarea funcțiilor de probabilitate ca funcții de apartenență	122
5.5.3.	Convexitatea grupărilor determinate de funcții de apartenență	125
5.5.4.	Separarea grupărilor	127
5.5.5.	Grupări compacte	134
5.6.	Sisteme interactive	135
5.7.	Observații bibliografice	142
ANEXA 1.	Eficacitatea strategiilor de selecție cu funcții de apropiere	147

ANEXA	2. Metode de rezolvare a ecuației fundamentale din analiza claselor latente	156
ANEXA	3. Teorema de separare a mulțimilor fuzzy	161
Bibliografie	163
Information Storage and Retrieval Systems (Abstract)	175

INTRODUCERE

Această monografie prezintă unele rezultate ale cercetării în domeniul sistemelor de informare.

Teoria sistemelor de informare se încadrează în domeniul mai larg al informaticii, care acoperă o mare varietate de topici, cum ar fi inteligența artificială și neurocibernetica, teoria automatelor și a mașinilor secvențiale, programarea automată, teoria deciziilor, sistemele de traducere automată, sistemele instruibile, recunoașterea formelor, simularea comportării omului și aplicarea calculatoarelor numerice pentru a rezolva probleme complexe militare, industriale, sau administrative. Informatica poate fi privită deci ca știința extinderii intelectului.

Una dintre cele mai remarcabile activități intelectuale o constituie comunicarea informației formulată în limbaj natural.

Informația transmisă prin limbaj natural poate fi folosită imediat sau poate fi destinată unei păstrări îndelungate. În ultimul caz, mijlocul principal de transmitere este indirect, static, prin documente scrise.

Pentru a fi transmise, informațiile sînt fixate pe documente scrise fără să se știe dacă, unde și de către cine vor fi cerute. La rîndul lor, utilizatorii nu știu dacă, unde și de către cine s-au scris documentele de care au nevoie.

Utilizatorii documentelor sînt separați de autorii lor prin spațiu, timp și limbă și de aceea transferul informației se face cu întârziere.

Este știut că, pentru a fi consultate, documentele sînt depozitate în colecții. Dacă colecția este sistematizată, atunci avem de a face cu o bibliotecă.

Accesul la informația stocată în bibliotecă a devenit din ce în ce mai greu pe măsură ce colecția și-a mărit volumul. După cel de-al doilea război mondial, numărul publicațiilor a crescut în mod exploziv, astfel încît bibliotecile își dublează volumul într-un interval de 10 ani, tinzînd să devină agenții din ce în ce mai ineficiente pentru

transferul informațiilor. Se pare că numărul total de cărți, opusculă, ziare etc. existente în toate bibliotecile din lume se ridică la peste 80 de milioane de titluri. Acest număr este în creștere în fiecare an cu mai mult de 3%. În fiecare minut în lume se editează circa 2 000 de pagini de tipar.

Problema principală a exploatării bibliotecilor este căutarea și extragerea din depozite. Pentru această căutare se pierde o cantitate colosală de timp și de muncă. Se întâmplă ca uneori să fie mai simplu să se efectueze o cercetare din nou, decât să se caute rezultatele unor cercetări similare în literatură.

În mod firesc a apărut necesitatea revizuirii metodelor de informare.

Progresele ultimei decade în domeniul prelucrării datelor și a copierii documentelor au condus la speranța că se poate mecaniza și procesul de informare.

Așa cum omul poate determina sensurile cuvintelor și propozițiilor pentru a putea face raționamente despre similitudinea sensurilor, redundanță, inconsistență sau relevanță, un sistem de informare poate face diferite operații asupra limbajului natural, cum ar fi analiza pentru detectarea și eliminarea informației redundante, selecția automată a informației relevante, verificarea automată a consistenței informației. Aceste sisteme prelucrează limbajul natural pe baza înțeleșului.

Cu câțiva ani în urmă a devenit limpede că memoria mare și posibilitatea de prelucrare rapidă a calculatoarelor electronice numerice le recomandă ca singura soluție pentru realizarea unor sisteme de informare satisfăcătoare.

Prin sistem de informare înțelegem un ansamblu de procedee pentru înregistrarea și selecția informațiilor. Ca urmare a actului de selecție un sistem de informare poate furniza direct informații sau poate furniza documente purtătoare de informații. În primul caz este vorba de sisteme care răspund la întrebări, iar în al doilea, de sisteme de regăsire a documentelor.

În cele ce urmează se va vorbi numai de sisteme de regăsire a documentelor.

Aceste sisteme înmagazinează documente și, ca răspuns la cereri de selecție, furnizează la ieșire adresele, rezumatele sau textul integral al documentelor.

Pentru a se putea face această operație, documentele nu sînt înmagazinate în formă originală, ci, ca urmare a unei analize, ele

sînt reprezentate prin termeni caracteristici pe care în continuare îi vom numi *descriptori*.

Astfel, un document este reprezentat în sistem printr-o *înregistrare*, care reprezintă suma descriptorilor săi și fiecărui sistem îi corespunde mulțimea descriptorilor folosiți pentru a reprezenta mulțimea documentelor.

Prin înregistrarea informațiilor înțelegem deci operația care constă în transcrierea descriptorilor într-o memorie.

Căutarea informațiilor înseamnă consultarea descriptorilor fiecărei înregistrări, pentru a selecționa pe acelea ce sînt apropiate de o submulțime dată a descriptorilor, numită cerere de selecție.

În general se cere să se selecționeze documente care în afara unor complicate combinații de descriptori trebuie să satisfacă și unele restricții. De exemplu, se poate cere ca sistemul să furnizeze o listă de articole care tratează despre „influența microstructurii asupra temperaturii de tranziție a superconductorilor intermetalici” cu restricțiile : toți autorii cu excepția acelor care lucrează în țara X, nimic înainte de anul 1962, nimic în japoneză, nimic din revista X, nu mai mult de 100 articole.



Problema cea mai importantă și cea mai dificilă într-un sistem de informare este cea a caracterizării documentelor, adică fixarea descriptorilor. Indexarea se poate face automat *) prin cîteva metode de analiză a conținutului cu grad de complexitate diferit. La baza acestor metode stă fie un criteriu statistic, fie un criteriu sintactic. În primul caz descriptorii sînt reprezentați de cuvintele cu anumită frecvență de apariție și folosind dicționare de interdicție pentru a elimina cuvintele uzuale. În al doilea caz, din text sînt selecționate propoziții referențiale ale căror predicate determină gradul de conexiune al unui descriptor la un document.

O a doua problemă importantă a sistemelor de informare o constituie strategiile de selecție, adică metodele de depistare a înregistrărilor al căror conținut este apropiat de cel al cererii de selecție.

Valoarea datelor păstrate în memoria sistemului depinde în mare măsură de rapiditatea cu care pot fi folosite. O a treia problemă care apare în proiectarea unui sistem de informare o constituie deci

*) Astăzi în unele reviste științifice, imediat după titlul unui articol este prezentat un șir de descriptori fixați de autor, care-l caracterizează conținutul.

organizarea stocului de înregistrări pentru ca timpul de răspuns să fie mic.

Din cele arătate mai sus rezultă că în sistemele de informare se pot separa trei procese distincte : un proces de analiză (indexare), un proces de organizare a colecției și un proces de confruntare a unei cereri de selecție cu colecția înmagazinată.

În legătură cu sistemele de înmagazinare și regăsire a informațiilor, a apărut un corp crescând de cunoștințe empirice și teoretice.

În această lucrare s-a efectuat mai ales un studiu al sistemelor de selecție, și al metodelor de organizare a colecțiilor de documente într-o încercare de a organiza într-un cadru logic unele rezultate ale cercetărilor efectuate în ultimii ani. Colectarea rezultatelor semnificative într-un domeniu nou, în plină dezvoltare, implică un anumit risc datorită probabilității de a trece cu vederea unele aspecte, însă se justifică prin necesitatea existenței unui cadru general în care să se poată evalua sistemele.

Prezenta lucrare este izvorâtă din punctul de vedere obișnuit al inginerului, care ar dori să aibă la dispoziție mijloace de proiectare.

În orice știință, cele două utilizări importante ale matematicii sînt formularea și soluționarea problemelor, pentru formulare fiind folosite modele. Noțiunea de model este aplicată în matematicile pure cînd se consideră un sistem formal, iar în matematicile aplicate cînd se consideră mărimi fizice, statice dacă natura lor este independentă de timp sau dinamice în caz contrar. Definind o structură ca o mulțime de mărimi statice și un proces ca o mulțime de mărimi dinamice, vom privi un sistem ca un proces. Construind un model se realizează un cadru general în care pot fi descrise toate sistemele existente sau se poate dezvolta un sistem mai generalizat. Poziția aceasta este cu totul deosebită de cea din fizică. În fizică există procese bine definite care pot fi descrise de modele. Unele rezultate ale procesului pot fi calculate cu ajutorul modelului și măsurate independent pe cale experimentală. Dacă cele două rezultate concordă, modelul este considerat valid. În domeniul informării nu există un astfel de sistem bine definit, ci, dimpotrivă, un sistem trebuie construit după un model. În acest caz trebuie găsit un model încît un sistem de informare proiectat conform acestui model să îndeplinească cît mai bine cerințele impuse anterior.

Toate încercările de a elabora modele ale sistemelor de informare s-au lovit de noțiunea de relevanță. În general prin relevanță se înțelege asemănarea dintre o înregistrare și o cerere de selecție.

Măsurarea în termeni cantitativi a acestei asemănări fiind dificilă, pînă nu de mult s-au considerat ca relevante toate documentele care cuprind integral descriptorii cererii. Astfel, procesul de selecție însemna o separare a înregistrărilor relevante de cele nerelevante.

În această lucrare se abordează altfel această problemă.

Procesul de selecție înseamnă aici ordonarea întregii colecții de înregistrări cu ajutorul unei funcții de mulțime numită funcție de selecție. Răspunsul sistemului nu mai este privit ca o dihotomie, ci ca graficul funcției de selecție.

Fără să se piardă generalitatea în întreaga lucrare, sînt tratate numai funcții de selecție cu valori în intervalul $[0, 1]$, cu toate că se pot construi funcții de selecție cu valori în toată dreapta reală sau în orice spațiu normat.

Definind strategia de selecție ca cuplul format din cererea de selecție și funcția de selecție, introducînd conceptul de răspuns comandat și aplicînd criteriul de evaluare a eficienței unei funcții de decizie folosit în teoria sistemelor instruibile, se demonstrează că o strategie de selecție este cu atît mai eficientă, cu cît funcția de selecție are mai mulți termeni. Pornind de la această constatare, în lucrare sînt introduse noi strategii cu funcții pătratice, demonstrînd că procesul de indexare poate fi privit ca un proces stohastic în care descriptorii sînt variabile aleatoare.

Impunînd ca o funcție de selecție să fie o restricție a unei măsuri exterioare, se pot construi funcții de selecție aditive. În acest caz prin procesul de selecție se poate realiza o repartitie de probabilități.

Lucrarea are următoarea organizare :

În capitolul 1 se prezintă unele rezultate generale ale teoriei indexării documentelor, care conduc la reprezentarea vectorială a înregistrărilor și la noțiunea de matrice de fixare, ale cărei elemente reprezintă valorile conexiunii descriptorilor la documente.

În capitolul 2 este prezentată o teorie a strategiilor de selecție bazată pe noțiunea de funcție de selecție, funcție ce introduce o relație de ordine nestrictă pe mulțimea înregistrărilor. Se dau două metode de evaluare a echivalenței strategiilor de selecție, una bazată pe rangul introdus prin ordonare de funcția de selecție și cealaltă pe măsura unor submulțimi care constituie răspunsul comandat al sistemului.

O cerere de selecție, liniară, nu presupune existența unei relații între descriptori, astfel încît formularea unei cereri cu descrip-

torii „aeronavă” și „producție” nu permite selectarea unei înregistrări cu descriptorii „avion” și „fabricație”. Pentru eliminarea acestui inconvenient există trei metode: introducerea în sistem a unui dicționar de sinonime, structurarea cererii de selecție sau selecția asociativă. În capitolul 2 sînt analizate cererile structurate, iar în capitolul 3 se prezintă o teorie a selecției prin asociere bazată pe matricea de similitudine între descriptorii sistemului, considerînd că similitudinea poate fi caracterizată de coapariția descriptorilor.

Capitolul 4 tratează problema organizării colecției pornind de la caracteristica principală a unui sistem de selecție și anume faptul că sistemul trebuie să furnizeze adresele înregistrărilor în funcție de conținutul lor și nu invers, ca în cazul sistemelor de prelucrare a datelor. De aceea pentru colecțiile mari de documente sistemele cu căutare secvențială se dovedesc nesatisfăcătoare și se preferă sistemele cu organizare în listă, sistemele cu organizare în dicționar, sistemele cu memorii asociative și sistemele cu clasificare automată.

Capitolul 5 se ocupă de sisteme cu clasificare automată. Un prim procedeu de clasificare este bazat pe matricea de similitudine între înregistrări. Acest procedeu, pentru a fi aplicabil practic, trebuie să pornească de la nuclee inițiale cît mai apropiate de grupările finale. În lucrare este prezentată o metodă, bazată pe vectorii proprii ai matricei de similitudine, pentru găsirea unor nuclee inițiale optime. Un alt procedeu de clasificare automată se bazează pe transformarea grafului, determinat de matricea de similitudine, într-o rețea unidimensională, prin a cărei reorganizare se obțin grupări suprapuse. O a treia metodă de clasificare se bazează pe proprietatea matricelor instruibile de a fi sensibile la distanța dintre înregistrări prototip și înregistrări cu care au fost instruite.

În final este analizată o metodă de clasificare bazată pe funcții de apartenență, fiecare grupare fiind considerată ca o mulțime *fuzzy*. Construcția funcțiilor de apartenență pornește de la o matrice ale cărei elemente sînt probabilitățile descriptorilor de a aparține unei anumite grupări. Această matrice se poate obține fie printr-o metodă de instruire plecînd de la eșantioane date pentru fiecare grupare, fie prin metoda analizei claselor latente, considerînd întreaga colecție. Sînt analizate astfel cîteva funcții de apartenență care au proprietatea de a conduce la grupări convexe. Pentru a determina gradul de separare a două grupări se aplică teorema lui Zadeh de separare a mulțimilor *fuzzy* convexe.

Capitolul se încheie cu o scurtă analiză a sistemelor interactive.

La sfârșitul fiecărui capitol se fac observații pe marginea materialelor folosite la elaborarea lucrării.

Terminologia și notațiile folosite în lucrare urmează în general pe cele folosite în literatură, cu toate că, fiind vorba de noțiuni și teorii foarte noi, nu s-a impus încă un mod de a le desemna.

Deși lucrarea se ocupă numai de sisteme de înmagazinare și regăsire a documentelor, ea este suficient de generală pentru ca să poată fi aplicată și la alte sisteme de informare. Astfel de sisteme sînt, de exemplu, cele încercate acum pentru diagnosticul medical. În general, atunci cînd pune un diagnostic, un medic notează simptomele pacientului și le compară cu simptomele asociate diverselor maladii. Ca urmare a acestei comparații, medicul selectează una sau mai multe maladii care explică comportarea pacientului. Acest procedeu poate conduce la un diagnostic fals, deoarece procesul de comparare, care stă la baza stabilirii diagnosticului, depinde de memoria medicului, fie cînd selectează simptomele pacientului, fie cînd consideră toate maladiile caracterizate de acele simptome. Dacă într-un sistem automat simptomele unor maladii sînt considerate ca descriptorii unor înregistrări, teoria selecției expusă în această lucrare se aplică fără corecții esențiale.

1 ÎNMAGAZINAREA DOCUMENTELOR

Majoritatea modelelor pentru sistemele de înmagazinare și regăsire a informațiilor sînt bazate pe teoria mulțimilor, pornind de la constatarea că procesul de regăsire presupune mulțimi de obiecte: pe de o parte mulțimi de descriptori și pe de altă parte mulțimi de documente. De fapt datele fundamentale ale teoriei acestor sisteme sînt furnizate de relațiile ce există între cele două mulțimi.

Modelul particular care convine unei anumite situații date depinde însă de mulți factori dintre care cei mai importanți sînt următorii :

- colecțiile de obiecte pot fi statice în sensul că pentru fiecare mulțime de obiecte există un complement bine definit sau din contra colecțiile se pot schimba în timp ;

- spațiul cererilor de selecție poate fi identic cu spațiul înregistrărilor, astfel că cererile sînt formulate cu aceiași descriptori folosiți pentru identificarea documentelor, sau din contra formularea cererilor poate să nu urmeze aceleași restricții aplicabile identificării documentelor ;

- între descriptorii documentelor și cei ai cererilor pot fi definite relații sau dimpotrivă fiecare descriptor poate fi independent de oricare alt descriptor.

În cele ce urmează, dacă nu se specifică altfel, se consideră colecții statice, o singură mulțime de descriptori și descriptori independenți. Se va arăta însă că pornind de la aceste criterii se poate elabora un model cu care să se poată interpreta și sisteme ce se abat de la condițiile impuse mai sus.

1.1. PROCESUL DE INDEXARE

Fie T mulțimea documentelor, D mulțimea descriptorilor, V mulțimea valorilor de conexiune a descriptorilor la documente,

S mulțimea intensităților de similitudine a descriptorilor. Cu ajutorul acestor mulțimi se pot construi propoziții primitive de forma

$$d(t) = v, \quad d \in D, \quad t \in T, \quad v \in V,$$

adică în limbaj neformal descriptorul d este atașat documentului t cu valoarea v .

Numim proces de indexare procesul stabilirii propozițiilor primitive.

Cu ajutorul propozițiilor primitive se pot construi următoarele propoziții de tip conjunctiv sau disjunctiv care se referă la atașarea descriptorilor la documente :

$$\text{Con } [d_j(t) = v_1, d_k(t) = v_2] = s_1,$$

adică în limbaj neformal descriptorii d_j și d_k ce apar în documentul t cu valori v_1 și v_2 sînt similari cu intensitatea de similitudine s_1 ;

$$\text{Dis } [d_j(t) = v_1, d_k(t) = v_2],$$

adică în limbaj neformal la documentul t este fixat fie descriptorul d_j , fie descriptorul d_k .

Metodele de indexare pot fi clasificate atît după modul de definire al propozițiilor de tip conjunctiv, cît și după modul de alegere al mulțimilor V și S . Dacă intensitatea de similitudine a descriptorilor poate fi dedusă din coapariția lor, atunci indexarea se numește fără indicatori de legătură. În acest caz numai propozițiile primitive pot fi axiome.

1.2. CRITERII DE INDEXARE AUTOMATĂ

1.2.1. Criteriul statistic

Fie mulțimea $D = \{d_1, d_2, \dots, d_n\}$. Un document t poate fi reprezentat ca o reuniune a mulțimilor disjuncte $\{d_i\}$ pe care le notăm Δ_i ,

$$t = \bigcup_{i=1}^n \Delta_i.$$

Mulțimea t este evident un trib deoarece

dacă $\Delta_i \in t$, atunci $\Delta_i - \Delta_i \in t$;

dacă (Δ_n) este un șir de mulțimi ale lui t , atunci reuniunea

$\bigcup_{n=1}^p \Delta_n$ a șirului aparține lui t ;

cel puțin una din mulțimile Δ_i este vidă.

Fie μ o măsură scalară reală pozitivă pe tribul t , adică o aplicație aditivă de mulțimi a tribului t în dreapta reală,

$$\mu : t \rightarrow R.$$

Atunci

$$\mu \left(\bigcup_{n=1}^p \Delta_n \right) = \sum_{n=1}^p \mu(\Delta_n),$$

oricare ar fi $p \in R$, și oricare ar fi șirul (Δ_n) de mulțimi disjuncte ale tribului t , ceea ce implică $\mu(\emptyset) = 0$.

Conform criteriului statistic, valoarea conexiunii dintre un descriptor d_i și un document t este valoarea măsurii mulțimii Δ_i atașată descriptorului d_i .

Un exemplu de măsură este numărul cardinal care satisface condiția

$$\text{card}(\bigcup \Delta_i) = \sum \text{card} \Delta_i$$

sau probabilitatea care satisface condițiile

$$\pi(\bigcup \Delta_i) = \sum \pi(\Delta_i),$$

$$\pi(t) = 1.$$

Funcția π poate fi

$$\pi(\Delta_i) = \frac{\text{card} \Delta_i}{\text{card} \bigcup_{i=1}^n \Delta_i}.$$

1.2.2. Criteriul sintactic

Criteriul sintactic impune identificarea expresiilor care ocupă o poziție referențială în propozițiile documentului. Acest criteriu

este aplicabil numai acelor propoziții în care este posibil să se izoleze cuvinte (sau grupuri de cuvinte) ce apar în poziții identificabile. Astfel de propoziții se numesc propoziții cu formă canonică. Din punct de vedere logic, cea mai importantă proprietate a unei propoziții cu formă canonică este că are structură de relație. În acest caz problema reprezentării unui document se reduce la problema logică a identificării acelor cuvinte ce reprezintă argumentele predicatelor în propoziții cu formă canonică.

Fie un predicat ireflexiv $P(d_1, d_2, \dots, d_k)$ cu k argumente. Fiecărui argument d_h i se asociază o mulțime finită D_h . Prin ipoteză

$$\text{card } D_h = k \text{ pentru toți } h.$$

Fiindcă P este un predicat ireflexiv, toate argumentele sale sînt diferite. Deci pentru toți i, j distincti $D_i \neq D_j$. Deoarece P stabilește același grad de conexiune pentru fiecare pereche distinctă a argumentelor sale, distanța minimă între două mulțimi trebuie să fie aceeași pentru fiecare pereche. Folosind ca distanță numărul cardinal al diferenței simetrice,

$$\delta(D_i, D_j) = \text{card } (D_i + D_j),$$

se observă că această distanță este minimă cînd $\text{card } (D_i + D_j) = 2$. Ținînd seama de identitățile

$$\text{card } (D_i \cup D_j) + \text{card } (D_i \cap D_j) = \text{card } D_i + \text{card } D_j,$$

$$\text{card } (D_i \cup D_j) - \text{card } (D_i \cap D_j) = \text{card } (D_i + D_j),$$

se poate scrie

$$\text{card } (D_i \cap D_j) = k - 1.$$

Fiindcă există C_k^n intersecții distincte a k mulțimi luate cîte n și numărul cardinal al fiecărei intersecții este $k - 1$,

$$\begin{aligned} \text{card } (D_1 \cup D_2 \cup \dots \cup D_k) &= C_k^1 \text{card } D_i - C_k^2 \text{card } (D_i \cap D_j) + \\ &+ C_k^3 \text{card } (D_i \cap D_j \cap D_k) - \dots \pm C_k^k \text{card } (D_1 \cap D_2 \cap \dots \cap D_k) = \\ &= k^2 - (k - 1)^2 = 2k - 1. \end{aligned}$$

Deci conexiunea dintre un descriptor și un predicat este măsurată de numărul cardinal al unei mulțimi finite astfel ca numărul cardinal al reuniunii tuturor mulțimilor să fie $2k - 1$, în timp ce numărul car-

dinal al intersecției oricăror două mulțimi este $k - 1$. Cu alte cuvinte, valoarea conexiunii unui descriptor la un predicat este k dacă predicatul are k locuri. Valoarea conexiunii unui descriptor d_i la un document este suma tuturor valorilor de conexiune dintre d_i și predicatele documentului care conțin pe d_i ca argument.

1.3. ÎNREGISTRAREA DOCUMENTELOR

1.3.1. Vectorul înregistrare

În urma procesului de indexare se obține o familie $(A_k) \subset \subset P(T)$ de părți disjuncte două câte două a căror reuniune este T ,

$$A_k = \{t \mid d(t) = v_k\}.$$

Deoarece (A_k) este o partiție a mulțimii T și v_k sînt numere reale, funcția d definită pentru orice $t \in T$ prin egalitatea

$$d(t) = v_k$$

este o variabilă aleatoare,

$$d : T \rightarrow V, \quad V \subset R.$$

Considerăm n variabile aleatoare d_1, d_2, \dots, d_n și aplicația

$$t \rightarrow (d_1(t), d_2(t), \dots, d_n(t))$$

a lui t în R^n . Atunci în spațiul n -dimensional fiecare document $t \in T$ este definit de vectorul

$$x = \{d_k(x) \mid k = 1, \dots, n\},$$

unde $d_k(x) = d_k(t)$, pe care îl vom numi vector înregistrare sau simplu înregistrare. Coordonatele vectorului corespund descriptorilor fixați prin axiome și valoarea fiecărei coordonate corespunde unui element al mulțimii V .

Înregistrarea x este o submulțime a mulțimii D , deoarece

$$d_k(x) = 0 \rightarrow d_k \notin x,$$

$$d_k(x) = 1 \rightarrow d_k \in x.$$

1.3.2. Matricea de fixare

Mulțimea X a vectorilor x formează o matrice F numită matrice de fixare:

$$F = \begin{bmatrix} d_1(x_1) & d_2(x_1) & \cdot & \cdot & \cdot & d_n(x_1) \\ d_1(x_2) & d_2(x_2) & \cdot & \cdot & \cdot & d_n(x_2) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ d_1(x_m) & d_2(x_m) & \cdot & \cdot & \cdot & d_n(x_m) \end{bmatrix}.$$

Coloanele matricei corespund descriptorilor sistemului, iar liniile matricei corespund înregistrărilor sistemului. Matricea se mai poate scrie

$$F = \begin{bmatrix} v_{11} & v_{21} & \cdot & \cdot & \cdot & v_{n1} \\ v_{12} & v_{22} & \cdot & \cdot & \cdot & v_{n2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ v_{1m} & v_{2m} & \cdot & \cdot & \cdot & v_{nm} \end{bmatrix},$$

unde v_{ij} este valoarea descriptorului d_i pentru o înregistrare x_j .

Matricea de fixare devine o matrice booleană dacă $V = \{0, 1\}$. În acest caz

$v_{ij} = 0$ dacă înregistrarea x_j nu are descriptorul d_i ;

$v_{ij} = 1$ dacă înregistrarea x_j are descriptorul d_i .

Matricea de fixare devine o matrice stohastică dacă elementele v_{ij} au fost determinate ca valori ale funcției π din § 1.2.1.

1.4. OBSERVAȚII BIBLIOGRAFICE

Cel mai cuprinzător studiu al problemelor indexării automate a fost făcut de Stevens [181], care a folosit o bibliografie cu peste 650 lucrări. La baza criteriului statistic stau lucrările lui Luhn [89], [90], care la IBM a făcut primele încercări de indexare automată. Aceste încercări s-au bazat pe presupunerea că cu cât apare mai des un cuvânt într-un document, cu atât mai probabil este ca acel cuvânt să fie un indicator semnificativ al subiectului acelui document. Există acum programe de numărare a cuvintelor într-un text astfel încât un calculator poate furniza o listă a tuturor cuvintelor din text aranjate în ordinea frecvenței de apariție. Cuvintele funcțio-

nale ca articolele, conjuncțiile, prepozițiile etc. sînt excluse folosind un dicționar de interdicție. În funcție de gradul de indexare dorit, cuvintele care au aceeași rădăcină pot fi numărate ca repetiții ale rădăcinii fixe sau separat. Prelucrarea statistică a textului furnizează deci o listă de cuvinte aranjate după frecvența de apariție. Din această listă, specificînd o valoare minimă a frecvenței, sînt selectați automat descriptorii.

O'Connor [131] a studiat problema frecvenței minime de selecție. Se crede acum că pentru a face eficace indexarea automată este necesar să se limiteze numărul descriptorilor eliminînd așa-numiții termeni nedescriptivi.

Cleverdon [23], de exemplu, este de părere că indexarea este bună selectînd toate cuvintele care apar în document mai mult de șase ori.

Edmunson și Wyllys [39] au împrumutat un principiu din teoria informației care sugera că semnificația unui cuvînt ar putea fi o funcție de raritate mai de grabă decît de frecvența de apariție. Ei au propus ca fiecărui cuvînt să i se calculeze un factor de semnificație $f - r$ sau f/r , unde f înseamnă frecvența de apariție a cuvîntului în document, iar r frecvența de folosire în general. Astfel un cuvînt folosit rar pentru scopuri ordinare care apare des într-un document va avea un factor de semnificație mare pentru acel document în timp ce cuvintele normal rare folosite rar în document și cuvintele comune folosite des vor primi valori mici.

Maron [97] și Lévery [86] au folosit doi parametri statistici pentru selecția descriptorilor: frecvența unui cuvînt și caracteristicile sale în diferite texte.

Kraveț, Moscovici și Șenderov [76] de la Institutul central de cercetări pentru informarea din brevete din Moscova lucrează cu o metodă de indexare bazată pe măsurarea deviației frecvenței cuvîntului de la frecvența medie teoretică. Mărimea acestei deviații dă valoarea de informare a descriptorului. Există un număr de cuvinte caracteristice la fiecare text și frecvența acestor cuvinte deviază de obicei de la frecvența normală în limbă. Astfel cuvîntul CAP este folosit mai frecvent decît cuvîntul DINTE, însă în textele de stomatologie proporția este inversată.

Simmons și McConlogue [171] au arătat că pentru sistemele cu răspuns la întrebări este indicată indexarea maximă, eliminînd numai un număr minim de cuvinte funcționale.

Climenson, Hardwick și Jacobson [24] consideră că pentru selectarea descriptorilor criteriul statistic nu este suficient și că indexarea trebuie să se bazeze pe recunoașterea și manipularea structurii sintactice a textului.

Baxendale [11], [12] a elaborat primul program pentru analiza sintactică. Criteriul sintactic din § 1.2.2 se datorește lui Hillman [63], care a folosit teoria simplității elaborată de Goodman [54]. Acest criteriu stă la baza sistemului automat de indexare al textelor elaborat de Hillman și Reed [65] la Lehigh University. Acest sistem este bazat pe o gramatică sensibilă la context și un dicționar cu trei sute de cuvinte-functor și sufixe. Fiecărui cuvânt dintr-un text i se fixează o categorie sintactică și i se identifică frazele referențiale care se segmentează în propoziții simple din punct de vedere sintactic. Această etapă este numită microcategorizare. În următoarea etapă, numită macrocategorizare, predicatele sînt izolate și se fixează valori argumentelor. Kasarda [70] a raportat în 1967 primele rezultate experimentale cu sistemul de la Lehigh University.

O teorie a limbajelor de indexare bazate pe predicate a fost schițată de Sanders [166].

2 STRATEGII DE SELECȚIE

Numim sistem de regăsire un sistem (X, D, V, S, γ) compus din patru mulțimi nevide și o funcție definită pe aceste mulțimi. Mulțimea X se numește mulțimea înregistrărilor din sistem, mulțimea D se numește mulțimea descriptorilor sistemului, mulțimea V se numește mulțimea valorilor descriptorilor, mulțimea S se numește mulțimea intensităților de similitudine a descriptorilor, iar funcția γ se numește funcția de selecție a sistemului.

Sistemele de regăsire pot fi clasificate după modul de alegere al mulțimilor D , V și S . Dacă intensitatea de similitudine a descriptorilor poate fi dedusă din coapariția lor, atunci sistemul se numește fără indicatori de legătură. Acesta este cazul sistemelor automate și de aceea în cele ce urmează vom vorbi numai de sisteme fără indicatori de legătură la care mulțimea S se deduce automat din celelalte mulțimi ale sistemului.

2.1. FUNCȚIA DE SELECȚIE

Funcția de selecție a sistemului stabilește o aplicație γ a mulțimii X pe dreapta reală R ,

$$\gamma : X \rightarrow R,$$

adică o corespondență prin care fiecărui element $x \in X$ i se asociază un element $\gamma(x) \in R$. Corespondența $x \rightarrow \gamma(x)$ este reprezentată prin perechi ordonate $(x, \gamma(x))$.

Numim răspuns al sistemului de selecție graficul funcției, adică mulțimea

$$\{(x, \gamma(x)) \mid x \in X\}.$$

Fie $r = \text{card } X$ și $I = \{1, 2, \dots, r\}$ o mulțime densă de întregi. O funcție de selecție γ induce pe mulțimea X o ordonare, adică o aplicație

$$\xi : X \rightarrow I.$$

Aplicația ξ este un epimorfism numai dacă funcția γ induce o ordonare totală pe mulțimea X .

2.2. FUNCȚIA DE SELECȚIE CA RESTRICȚIE A MĂSURII EXTERIOARE

Considerăm mulțimea D și mulțimea $P(D)$ a tuturor submulțimilor mulțimii D . Mulțimea $P(D)$ este un trib deoarece are proprietățile

$$A \in P(D), B \in P(D) \rightarrow A - B \in P(D),$$

$$(A_k) \in P(D) \rightarrow \bigcup A_k \in P(D),$$

$$D \in P(D),$$

$$\emptyset \in P(D).$$

Măsura exterioară $\mu^* : P(D) \rightarrow R$ este o funcție reală, definită pe $P(D)$ cu următoarele proprietăți :

$$\mu^*(\emptyset) = 0;$$

μ^* este numerabil subaditivă, adică pentru orice șir (A_k) de mulțimi din $P(D)$ disjuncte sau nu avem

$$\mu^*(\bigcup A_k) \leq \sum \mu^*(A_k);$$

μ^* este monotonă, adică dacă $A \in P(D)$, $B \in P(D)$ și $A \subseteq B$, atunci $\mu^*(A) \leq \mu^*(B)$.

Deoarece $X \subset P(D)$, atunci dacă

$$x \in X \rightarrow \mu^*(x) = \gamma(x),$$

funcția γ este restricția lui μ^* la X . Din implicația de mai sus rezultă că o funcție dată μ^* are o singură restricție la mulțimea X dată.

2.3. FUNCȚIA DE SELECTIE CA FUNCȚIE ETAJATĂ

Aplicația $\gamma: X \rightarrow R$ este etajată pe $P(X)$, deoarece este o aplicație simplă, adică are un număr finit de valori v_1, v_2, \dots, v_g , $g \leq r$, mulțimile

$$X_k = \{x | \gamma(x) = v_k\} \quad k = 1, 2, \dots, g$$

apartțin familiei $P(X)$ care este un trib.

Considerăm funcția caracteristică a mulțimii X_k , adică aplicația

$$\chi_{X_k}: X \rightarrow R,$$

definită pentru fiecare element $x \in X$ astfel:

$$\chi_{X_k}(x) = \begin{cases} 1 & \text{dacă } x \in X_k, \\ 0 & \text{dacă } x \notin X_k. \end{cases}$$

Deoarece γ este o funcție finită etajată pe $P(X)$, ea se poate scrie sub forma

$$\gamma = \sum_{k=1}^g v_k \chi_{X_k}.$$

Mulțimea aplicațiilor etajate pe $P(X)$ formează un spațiu vectorial pe corpul numerelor reale. Fie Γ această mulțime $\gamma_1 \in \Gamma$, $\gamma_2 \in \Gamma$ și $c \in R$. Atunci

$$c \gamma_1 \in \Gamma,$$

$$\gamma_1 + \gamma_2 \in \Gamma.$$

2.4. CONSTRUCȚIA FUNCȚIEI DE SELECTIE

Pentru o înregistrare

$$x = \{d_k(x) | k = 1, \dots, n\},$$

funcția γ ia valoarea

$$\gamma(x) = \gamma(d_k(x), c_1, c_2, \dots, c_n),$$

unde c_1, \dots, c_n sînt parametrii funcției.

O funcție de selecție poate fi scrisă sub forma

$$\gamma = \sum_{i=1}^n c_i \psi_i,$$

unde ψ_i sînt funcții reale uniforme liniar independente.

Spunem că γ este o funcție liniară dacă

$$\psi_i(x) = d_k(x), \quad k = 1, \dots, n.$$

Evident, o funcție liniară are n componente.

Spunem că γ este o funcție polinomială de ordinul r dacă $\varphi_i(x)$ este de forma $d_{k_1}^{a_1}(x) d_{k_2}^{a_2}(x) \dots d_{k_r}^{a_r}(x)$, unde $k_1, k_2, \dots, k_r = 1, \dots, n$ și $a_1, a_2, \dots, a_r = 0$ și 1.

O astfel de funcție are $h = \sum_{i=1}^r C_{n+i-1}^i$ componente. Dacă $r=2$ spunem că γ este o funcție pătratică.

O funcție pătratică are forma

$$\gamma(x) = \sum_{k=1}^n c_{kk} d_k^2(x) + \sum_{k=1}^{n-1} \sum_{j=k+1}^n c_{jk} d_j(x) d_k(x) + \sum_{k=1}^n d_k(x).$$

Această funcție are

n termeni $d_k^2(x)$,

n termeni $d_k(x)$,

$n(n-1)/2$ termeni $d_j(x) d_k(x)$,

Funcția poate fi pusă într-o formă matricială folosind notațiile

$$\begin{aligned} A &= (a_{jk}), \\ a_{jj} &= c_{jj}, \quad j = 1, \dots, n, \\ a_{jk} &= \frac{1}{2} c_{jk}, \quad j, k = 1, \dots, n, \quad j \neq k, \\ B &= \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}. \end{aligned}$$

În acest caz

$$\gamma(x) = x^t A x + x^t B.$$

2.5. STRATEGIA DE SELECȚIE

Specificarea parametrilor c ai funcției de selecție înseamnă specificarea unei submulțimi $q \in D$ pe care o vom numi cerere de selecție.

Numim proces de formulare a cererii de selecție procesul stabilirii propozițiilor de forma

$$d(q) = v, \quad d \in D, \quad v \in V.$$

Ca și în cazul unei înregistrări, o cerere de selecție poate fi reprezentată de un vector

$$q = \{d_k(q) \mid k = 1, \dots, n\}.$$

Numim strategie de selecție cuplul $\sigma = (q, \gamma)$.

Precizarea unei strategii de selecție înseamnă precizarea unei funcții de selecție γ_q astfel ca

$$\gamma_q(x) = \gamma(x, q).$$

Fie Q mulțimea cererilor de selecție. Vom spune că o familie de aplicații $\gamma_q : X \rightarrow R$, $q \in Q$, este parametrizată de mulțimea Q dacă

aplicația $\gamma : X \times Q \rightarrow R$, definită de

$$\gamma(x, q) = \gamma_*(x), \quad x \in X, q \in Q,$$

este continuă.

2.5.1. Strategii cu funcții aditive

Vom considera un clan de părți ale mulțimii D , generat de mulțimea X , adică o clasă nevidă \mathcal{C} de părți ale mulțimii D dacă

$$A - B \in \mathcal{C} \text{ oricare ar fi } A, B \in \mathcal{C},$$

$$A \cup B \in \mathcal{C} \text{ oricare ar fi } A, B \in \mathcal{C}$$

și π o funcție reală pozitivă definită pe \mathcal{C} care are proprietățile

$$\pi(D) = 1,$$

$$\pi(\bigcup A_k) = \sum \pi(A_k),$$

dacă $(A_k) \cup \mathcal{C}$ este o familie numerabilă de elemente din \mathcal{C} și dacă $\bigcup A_k \in \mathcal{C}$.

Pentru orice mulțime $A \in P(D)$

$$\mu^*(A) = \inf \sum \pi(A_k),$$

marginea inferioară luându-se pentru toate șirurile (A_k) de mulțimi din \mathcal{C} cu $A \subset \bigcup A_k$ și

$$\mu^*(A) = \pi(A).$$

Funcția π are proprietățile

$$A \in P(D), B \in P(D), B \supset A \rightarrow \pi(B - A) = \pi(B) - \pi(A),$$

$$A \in P(D), B \in P(D), B \supset A \rightarrow \pi(B) > \pi(A),$$

$$\pi(\emptyset) = 0,$$

$$0 \leq \pi(A) \leq 1,$$

$$\pi(\mathcal{C}A) = 1 - \pi(A).$$

Sistemul $\{D, P(D), \pi\}$ este un câmp de probabilitate complet aditiv.

Vom spune că o cerere de selecție q definește o repartiție de probabilități. Răspunsul sistemului va fi atunci

$$\{(x, \pi(x)) \mid x \in X\}.$$

Unei strategii de selecție $\sigma = (q, \pi)$ îi corespunde o funcție de selecție astfel ca

$$\pi(x, q) = \pi_q(x).$$

Un exemplu de funcție aditivă îl constituie funcția Reitsma-Sagalyn

$$\pi_{RS} = \frac{\sum_{k=1}^n \frac{\min(d_k(x), d_k(q))}{\max(d_k(x), d_k(q))}}{n},$$

care pentru sisteme cu $V = \{0, 1\}$ devine

$$\frac{\sum_{k=1}^n d_k(x) d_k(q)}{n}.$$

2.5.2. Strategii cu funcții de apropiere

Numim funcție de apropiere peste mulțimea $X \times Q$ o funcție α care stabilește o aplicație

$$\alpha : X \times Q \rightarrow R$$

ce fixează fiecărei perechi (x, q) de elemente din $X \times Q$ un număr real astfel ca

$$[\forall x \in X] (\alpha(x, q) = \alpha(q, x)),$$

$$[\forall x \in X] (x = q \Leftrightarrow \alpha(x, q) = 1).$$

În cele ce urmează sînt analizate cîteva funcții de apropiere. Funcția cosinus :

$$\alpha_c = \frac{\sum_{k=1}^n d_k(x) d_k(q)}{\left(\sum_{k=1}^n d_k^2(x) \sum_{k=1}^n d_k^2(q) \right)^{1/2}}.$$

Această funcție măsoară unghiul dintre vectorii x și q . Fiindcă numitorul este produsul lungimilor vectorilor în spațiul n -dimensional, funcția va crește cu creșterea lungimii vectorilor. Produsul scalar al acestor vectori va crește în aceeași măsură sau mai puțin decât numitorul deoarece conform inegalității Cauchy-Buniakovski

$$\left(\sum_{k=1}^n d_k(x) d_k(q) \right)^2 \leq \sum_{k=1}^n d_k^2(x) \sum_{k=1}^n d_k^2(q).$$

Funcția va depinde deci de lungimea vectorilor.

Funcția hipersinus :

$$\alpha_{\text{HS}} = \frac{\sum_{k=1}^n d_k(x) d_k(q) d_k(q)}{\left(\sum_{k=1}^n d_k^2(x) d_k(q) \sum_{k=1}^n d_k^2(q) d_k(q) \right)^{1/2}}.$$

Efectul numeric al factorului suplimentar $d_k(q)$ este nul pentru că el poate fi simplificat. Acest factor este introdus pentru a reduce lungimea vectorului înregistrare, deoarece produsul $d_k^2(x) d_k(q)$ este zero când $d_k(x) > 0$, dar $d_k(q) = 0$. Produsul este diferit de zero numai când descriptorul d_k apare simultan în x și q . Cu alte cuvinte lungimea vectorului x este calculată în subspațiul spațiului Q . Fiindcă de obicei vectorul x este mai lung decât vectorul q , α_{HS} reduce dependența de lungime.

Funcția lui Parker-Rhodes-Needham :

$$\alpha_{\text{PRN}} = \frac{\sum_{k=1}^n d_k(x) d_k(q)}{\sum_{k=1}^n d_k^2(x) + \sum_{k=1}^n d_k^2(q) - \sum_{k=1}^n d_k(x) d_k(q)}.$$

Numitorul reprezintă suma produselor scalare maxime ale celor doi vectori în cazul corelării perfecte minus produsul scalar real. Diferența va fi totdeauna mai mare sau egală cu produsul scalar real.

Funcția lui Maron-Kuhns :

$$\alpha_{MK} = \frac{1 + \frac{\sum_{k=1}^n d_k(x) d_k(q) \sum_{k=1}^n \bar{d}_k(x) \bar{d}_k(q) - \sum_{k=1}^n d_k(x) \bar{d}_k(q) \sum_{k=1}^n \bar{d}_k(x) d_k(q)}{\sum_{k=1}^n d_k(x) d_k(q) \sum_{k=1}^n \bar{d}_k(x) \bar{d}_k(q) + \sum_{k=1}^n d_k(x) \bar{d}_k(q) \sum_{k=1}^n \bar{d}_k(x) d_k(q)}}{2}.$$

În această funcție \bar{d}_k este complementul lui d_k . În cazul vectorilor binari dacă $d_k(x) = 1$, atunci $\bar{d}_k(x) = 0$. În cazul vectorilor ale căror componente pot lua mai multe valori,

$$d_k(x) = v,$$

$$\bar{d}_k(x) = v_{\max} - v,$$

unde v_{\max} este valoarea maximă pe care o poate lua descriptorul în sistem. Deoarece în situațiile reale majoritatea componentelor sînt nule, nu se completează decît componentele nenule.

Funcția de suprapunere :

$$\alpha_S = \frac{\sum_{k=1}^n \min(d_k(x), d_k(q))}{\min\left(\sum_{k=1}^n d_k(x), \sum_{k=1}^n d_k(q)\right)}.$$

Numitorul este dat de vectorul cu cea mai mică sumă a componentelor. Numărătorul este un vector în spațiul înregistrărilor ale cărui componente aparțin vectorilor x și q .

Funcția minimax :

$$\alpha_{MM} = \frac{\sum_{k=1}^n \min(d_k(x), d_k(q))}{\sum_{k=1}^n \max(d_k(x), d_k(q))}.$$

Atît numărătorul cît și numitorul sînt vectori în spațiul înregistrărilor cu componente ce aparțin vectorilor x și q .

Funcția de suprapunere modificată :

$$\alpha_{SM} = \frac{\sum_{k=1}^n \bar{d}_k(x) \bar{d}_k(q)}{\sum_{k=1}^n \bar{d}_k(x)} .$$

Numitorul reprezintă suma componentelor vectorului înregistrare. Această sumă este totdeauna mai mare decât produsul scalar al vectorilor x și q dacă vectorul q este un vector binar.

În cazul când vectorii x și q sînt binari vom face următoarele notații :

$n_x = \sum \{ \bar{d}_k(x) \mid k = 1, \dots, n \}$ numărul de componente 1 în vectorul x ,

$n_q = \sum \{ \bar{d}_k(q) \mid k = 1, \dots, n \}$ numărul de componente 1 în vectorul q ,

$n_{xq} = \sum \{ \bar{d}_k(x) \bar{d}_k(q) \mid k = 1, \dots, n \}$ numărul de componente 1 comune în vectorii x și q ,

$\bar{n}_{xq} = \sum \{ \bar{\bar{d}}_k(x) \bar{\bar{d}}_k(q) \mid k = 1, \dots, n \}$ numărul de componente 0 comune în vectorii x și q .

Vectorii fiind binari este evident că

$$\sum \{ \bar{d}_k(x) \mid k = 1, \dots, n \} = \sum \{ \bar{d}_k^2(x) \mid k = 1, \dots, n \}.$$

Cu aceste notații se poate scrie

$$\alpha_C = \frac{n_{xq}}{(n_x n_q)^{1/2}} .$$

$$\alpha_{HS} = \frac{n_{xq}}{(n_{xq} n_q)^{1/2}} ,$$

$$\alpha_{PRN} = \frac{n_{xq}}{n_x + n_q - n_{xq}} ,$$

$$\alpha_S = \frac{n_{xq}}{\min(n_x, n_q)},$$

$$\alpha_{MM} = \frac{n_{xq}}{n_x + n_q - n_{xq}},$$

$$\alpha_{SM} = \frac{n_{xq}}{n_x}.$$

Se observă că în cazul vectorilor binari funcția α_{MM} este identică cu funcția α_{PRN} .

2.5.3. Strategii cu funcții de repartiție

În urma procesului de indexare se obține o familie (X_k) de părți disjuncte două câte două, a căror reuniune este mulțimea X a înregistrărilor :

$$X_k = \{x | d(x) = v_k\}.$$

Deoarece (X_k) este o partiție a mulțimii X și v_k sînt numere reale, funcția d definită pentru orice $x \in X$ prin egalitatea

$$d(x) = v_k$$

este o variabilă aleatoare :

$$d : X \rightarrow V, \quad V \subset R.$$

Considerăm n variabile aleatoare d_1, d_2, \dots, d_n și aplicația

$$x \mapsto (d_1(x), d_2(x), \dots, d_n(x))$$

a lui x în R^n .

Fie

$$I = \{(v_1, v_2, \dots, v_n) | v_1 < a_1, v_2 < a_2, \dots, v_n < a_n\}$$

și

$$X_{a_1, a_2, \dots, a_n} = \{x | d(x) \in I\}.$$

Funcția

$$F(a_1, a_2, \dots, a_n) = p(X_{a_1, a_2, \dots, a_n})$$

este funcția de repartiție a variabilelor aleatoare d_1, d_2, \dots, d_n .

O funcție normală F este definită prin densitatea de repartiție

$$f(v_1, v_2, \dots, v_n) = \frac{1}{(2\pi)^{n/2} |A|^{1/2}} e^{-\frac{1}{2} g(v_1, v_2, \dots, v_n)},$$

unde

$$g(v_1, v_2, \dots, v_n) = \sum_{i,j=1}^n a_{ij} v_i v_j$$

este o formă pătratică strict pozitivă, iar coeficienții a_{ij} sînt elementele unei matrice

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix}.$$

Considerăm funcția

$$\omega_N(x) = \frac{1}{(2\pi)^{n/2} |A|^{1/2}} e^{-\frac{1}{2} (x-q)^T A^{-1} (x-q)},$$

unde

$$x = \begin{bmatrix} d_1(x) \\ d_2(x) \\ \vdots \\ d_n(x) \end{bmatrix} \quad q = \begin{bmatrix} d_1(q) \\ d_2(q) \\ \vdots \\ d_n(q) \end{bmatrix}$$

și A este matricea dispersiei.

Hipersuprafețele $\omega_N(x) = c$ sînt hiperelipsoizi centrați pe punctul q . Acești hiperelipsoizi sînt suprafețe de egală probabilitate în spațiul n -dimensional al înregistrărilor. Răspunsul sistemului la o strategie cu funcție normală va fi deci o grupare elipsoidală centrată în jurul cererii de selecție.

Fiindcă A este pozitiv definit putem scrie

$$A^{-1} = T D D^t T^t,$$

unde T este o matrice $n \times n$ a vectorilor proprii ai matricei A^{-1} , iar DD^t este o matrice diagonală $n \times n$ ale cărei elemente sînt valorile proprii ale matricei A^{-1} . Cu transformările

$$z = \lambda x,$$

$$\lambda = D^t T^t,$$

funcția devine

$$\omega_N(x) = \frac{1}{(2\pi)^{n/2} |A|^{1/2}} e^{-\frac{1}{2} (x-\lambda a)^t (x-\lambda a)}.$$

Transformarea λ schimbă gruparea elipsoidală într-o grupare sferică.

Funcții de selecție pot fi construite și cu ajutorul distribuțiilor Pearson. Astfel funcția

$$\omega_{p2}(x) = \frac{\Gamma(m + n/2 + 1)}{\pi^{n/2} \Gamma(m + 1)} |W|^{1/2} (1 - (x - q)^t W(x - q))^m$$

reprezintă distribuția Pearson de tip II

$$\omega_{p2}(x) = \begin{cases} \omega_{p2}(x) & \text{peste regiunea } T, \\ 0 & \text{în afara regiunii } T, \end{cases}$$

unde T este interiorul hiperelipsoidului $(x - q)^t W(x - q) = 1$, $m \geq 0$, și

$$W = \frac{1}{2m + n + 2} A^{-1}.$$

Γ este funcția euleriană de a doua speță; pentru n întreg și pozitiv $\Gamma(n) = (n - 1)!$.

Funcția

$$\omega_{p7}(x) = \frac{\Gamma(m)}{\pi^{n/2} \Gamma(m - n/2)} |W|^{1/2} (1 + (x - q)^t W(x - q))^{-m},$$

unde $2m > n$, reprezintă distribuția Pearson de tip VII.

În particular, dacă $2m > n + 2$, matricea covariantă există și

$$W = \frac{1}{2m - n - 2} A^{-1}.$$

Ambele tipuri de distribuție Pearson tind spre distribuția normală când m tinde spre infinit.

2.6. ECHIVALENȚA STRATEGIILOR DE SELECȚIE

O strategie de selecție σ_i induce pe mulțimea X o ordonare astfel încât există o aplicație ξ din X în I . Creșterea rangului în mulțimea întregilor I reflectă descreșterea valorilor funcției de selecție*).

Spunem că două strategii sînt echivalente dacă induc aceeași ordonare pe mulțimea X , adică dacă conduc la același răspuns.

2.6.1. Deosebirea de răspuns

Numim răspuns comandat al sistemului de selecție mulțimea

$$X_k = \{x | \gamma_r(x) > c_k\}.$$

Notăm r_i rangul unei înregistrări $x_i \in X$ obținut ca urmare a aplicației ξ . Atunci

$$\xi_j(x_i) = r_{ji},$$

$$\xi_k(x_i) = r_{ki},$$

$$r_{i+1} > r_i.$$

Definim o funcție de răspuns

$$g(z) = \frac{1}{m_k} \sum_{i=1}^{m_k} \delta(z - r_i),$$

unde $m_k = \text{card } X_k$, iar δ este funcția treaptă unitate.

* Există și posibilitatea ca situația să fie inversată, ca de exemplu în cazul folosirii funcțiilor de distanță drept funcții de selecție.

Astfel pentru aplicația ξ_j ,

$$g_{\xi_j}(z) = \frac{1}{m_k} \sum_{i=1}^{m_k} \delta(z - r_{ji}),$$

iar pentru aplicația ξ_k

$$g_{\xi_k}(z) = \frac{1}{m_k} \sum_{i=1}^{m_k} \delta(z - r_{ki}).$$

Definim deosebirea de răspuns

$$\begin{aligned} \Delta_{jk} &= \int_1^r (g_{\xi_j}(z) - g_{\xi_k}(z)) dz = \\ &= \frac{1}{m_k} \sum_{i=1}^{m_k} \int_1^r (\delta(z - r_{ji}) - \delta(z - r_{ki})) dz = \\ &= \frac{1}{m_k} \left(\sum_{i=1}^{m_k} r_{ki} - \sum_{i=1}^{m_k} r_{ji} \right). \end{aligned}$$

Condiția de echivalență a strategiilor de selecție poate fi reformulată astfel: două strategii sînt echivalente dacă deosebirea de răspuns este nulă.

Luînd ca etalon o strategie $\sigma_1 = (q, \gamma_1)$ valorile Δ permit evaluarea sistemului, adică aprecierea proporției de elemente utile selecționate de sistem pentru fiecare din strategiile $\sigma_2, \sigma_3, \dots, \sigma_n$.

2.6.2. Lungimea de selecție

Fie o strategie de selecție σ_i care induce o ordonare a mulțimii X . Răspunsul comandat al sistemului este

$$X_{k,i} = \{x \mid \gamma_i(x) > c_k\}.$$

O altă strategie de selecție σ_j induce o altă ordonare a mulțimii X . În acest caz răspunsul comandat al sistemului este

$$X_{k,j} = \{x \mid \gamma_j(x) > c_k\}.$$

În mulțimea ordonată de strategia σ_j există o submulțime

$$X_{j,i} = \{x \mid \gamma_i(x) > c_i\}$$

astfel ca

$$X_{ki} \subset X_{ji}.$$

Considerăm o măsură $\mu: P(X) \rightarrow R$ și fie $X_{ji} = X_{ki} \cup X_j$. Atunci

$$\mu(X_{ji}) = \mu(X_{ki}) + \mu(X_j).$$

În mod similar pentru $X_{li} = X_{ki} \cup X_L$

$$\mu(X_{li}) = \mu(X_{ki}) + \mu(X_L).$$

Dacă drept măsură folosim numărul cardinal și definim lungimea de selecție ca numărul elementelor cu $\gamma_i(x) < c_k$ ce apar într-un răspuns comandat de (σ_i, c_k) , atunci condiția de echivalență a strategiilor de selecție poate fi reformulată astfel: două strategii sînt echivalente dacă conduc la aceeași lungime de selecție.

2.7. EFICACITATEA STRATEGIILOR DE SELECȚIE

Considerăm răspunsul comandat

$$X_k = \{x | \gamma_k(x) > c_k\}.$$

În acest caz, ca urmare a procesului de selecție se produce o dihotomie, adică o partiție a mulțimii X în două submulțimi disjuncte X_k și $\complement X_k$, prin hipersuprafața $\gamma_k(x) = c_k$.

Presupunem că în sistem sînt r înregistrări. Este evident că există 2^r dihotomii distincte, fiecare înregistrare putînd fi fixată la X_k sau la $\complement X_k$.

O măsură a eficacității unei strategii de selecție este numărul total de dihotomii pe care le poate efectua.

Dacă pozițiile a r înregistrări satisfac unele condiții, numărul dihotomiilor care pot fi realizate de o strategie $\sigma = (q, \gamma)$ va depinde numai de numărul de înregistrări r și de numărul de parametri ai funcției γ și nu de configurația înregistrărilor sau de forma funcției.

Pentru $r > n$ spunem că o mulțime de r puncte este în poziție generală într-un spațiu n -dimensional dacă și numai dacă nici o submulțime de $n + 1$ puncte nu stă pe un hiperplan $(n - 1)$ -dimensional. Cînd $r \leq n$, o mulțime de r puncte este în poziție generală dacă nici un hiperplan $(r - 2)$ -dimensional nu conține mulțimea.

În anumite cazuri speciale, importante în practică, înregistrările pot să nu fie în poziție generală. De exemplu dacă componentele înregistrării sînt binare, înregistrările sînt vîrfurile unui hipercub. În acest caz poziția generală implică că nici o submulțime de $n + 1$ vîrfuri nu poate sta pe aceeași față $(n-1)$ -dimensională.

Chiar cînd înregistrările nu sînt în poziție generală, expresia care se obține dă o margine superioară pentru numărul dihotomiilor.

Fie $L(r, n)$ numărul de dihotomii a r înregistrări obținute cu o funcție de selecție liniară, adică dublul numărului de moduri în care r puncte pot fi împărțite de un hiperplan $(n-1)$ -dimensional, considerînd că pentru fiecare împărțire sînt două clasificări diferite. Expresia generală pentru $L(r, n)$ se obține cu relația de recurență cunoscută

$$L(r, n) = L(r-1, n) + L(r-1, n-1).$$

Folosind condițiile la limită evidente

$$L(1, n) = 2,$$

$$L(r, 1) = 2r,$$

este ușor de verificat că

$$L(r, n) = \begin{cases} 2 \sum_{i=0}^n C_{r-1}^i & \text{pentru } r > n, \\ 2^r & \text{pentru } r \leq n. \end{cases}$$

Funcția γ este de forma $\sum_{i=1}^h c_i \psi_i$ și generează hipersuprafețe

pe care le vom numi hipersuprafețe γ . Pentru a calcula numărul dihotomiilor realizate de hipersuprafața γ observăm că fiecărui punct $x \in X$ îi corespunde un punct $g \in G$

$$g = \{\psi_i(x) \mid i = 1, \dots, h\}.$$

Deci mulțimii X cu r puncte în poziție generală în spațiul n -dimensional îi corespunde mulțimea G cu r puncte în spațiul h -dimensional. Fiindcă o dihotomie liniară a mulțimii G corespunde unei dihotomii γ a mulțimii X , atunci

$$\gamma(r, h) = \begin{cases} 2 \sum_{i=0}^h C_{r-1}^i & \text{pentru } r > h, \\ 2^r & \text{pentru } r \leq h. \end{cases}$$

Pentru o funcție pătratică care are $h = (n)(n + 3)/2$ componente

$$\gamma(r, h) = L\left(r, \frac{n(n + 3)}{2}\right).$$

Astfel o hipercuadrică este o suprafață de decizie mai puternică decât hiperplanul și putem spune că o strategie de selecție cu funcție pătratică va fi mai eficientă decât o strategie cu funcție liniară. Numărul componentelor unei funcții de selecție parametrizată depinde însă de numărul de componente nenule al vectorului care reprezintă cererea de selecție și deci implicit eficacitatea unei strategii de selecție va depinde de lungimea vectorului cerere.

În cele ce urmează se face o analiză comparativă a câtorva strategii tipice.

În cazul strategiei cu funcția π_{RS} , când funcția are valoarea c , ecuația

$$\sum_{k=1}^n d_k(x) d_k(q) = cn$$

reprezintă un hiperplan.

Funcția π_{RS} are deci n componente numai atunci când vectorul cerere de selecție are toate componentele nenule. De obicei numărul zerourilor este mare și eficacitatea acestei funcții este scăzută, când înregistrările sînt reprezentate prin vectori binari.

În cazul strategiei cu funcția α_{SM} când funcția are valoarea c ecuația

$$\sum_{k=1}^n d_k(x) d_k(q) - c \sum_{k=1}^n d_k(x) = 0$$

reprezintă de asemenea un hiperplan. Funcția α_{SM} are însă totdeauna n componente independent de numărul de zerouri în vectorul cerere, deoarece cel de-al doilea termen nu este influențat de acest vector.

În cazul strategiei cu funcția α_{PRN} când funcția are valoarea c ecuația

$$\sum_{k=1}^n d_k(x) d_k(q) - c \left(\sum_{k=1}^n d_k^2(x) + \sum_{k=1}^n d_k^2(q) - \sum_{k=1}^n d_k(x) d_k(q) \right) = 0$$

reprezintă o hipercuadrică. Funcția α_{PRN} are cel mult $2n$ componente.

În cazul strategiei cu funcția cosinus, cînd funcția are valoarea c , ecuația

$$\left(\sum_{k=1}^n d_k(\omega) d_k(q) \right)^2 - c \sum_{k=1}^n d_k^2(\omega) \sum_{k=1}^n d_k^2(q) = 0$$

reprezintă de asemenea a hiperquadrică. Funcția α_c are $n(n+1)/2$ componente.

Eficacitatea acestor ultime două funcții pătratic diferă sensibil numai pentru sisteme cu număr mare de descriptori.

În anexa 1 criteriul de apreciere a eficienței unei strategii de selecție este verificat practic pentru o colecție artificială de înregistrări.

2.8. RĂSPUNSUL SISTEMULUI

În cele ce urmează vom propune că funcția de selecție ia valori numai în intervalul $[0, 1]$.

Un răspuns X_q la o strategie de selecție $\sigma = (q, \gamma)$ este o mulțime de elemente $x \in X$ determinată de o funcție de selecție γ_q , care asociază cu fiecare $x \in X$ un număr real în intervalul $[0, 1]$, valorile $\gamma_q(x)$ reprezentînd gradul de apartenență al înregistrării x la răspunsul X_q . Astfel cu cît $\gamma_q(x)$ este mai aproape de unitate, cu atît este mai mare gradul de apartenență al elementului x la mulțimea X_q .

Un răspuns X_q este vid dacă și numai dacă

$$[\forall x \in X] (\gamma_q(x) = 0).$$

Două răspunsuri X_{q1} și X_{q2} sînt egale dacă și numai dacă

$$[\forall x \in X] (\gamma_{q1}(x) = \gamma_{q2}(x)).$$

Spunem că răspunsul X_{q1} este conținut în răspunsul X_{q2} adică $X_{q1} \subset X_{q2}$, dacă și numai dacă

$$[\forall x \in X] (\gamma_{q1}(x) \leq \gamma_{q2}(x)).$$

Pentru z funcții de selecție vor fi z strategii de selecție

$$\sigma_1 = (\gamma_1, q),$$

$$\sigma_2 = (\gamma_2, q);$$

$$\dots$$

$$\sigma_z = (\gamma_z, q).$$

Fiecare strategie σ_i determină un răspuns X_i .

Definim reuniunea a două răspunsuri X_i și X_j cu funcțiile de selecție γ_i și γ_j , ca fiind un răspuns $X_{ij} = X_i \cup X_j$ la care

$$[\forall x \in X] (\gamma_{ij}(x) = \max(\gamma_i(x), \gamma_j(x))),$$

adică cel mai mic răspuns care conține și X_i și X_j .

Dacă X_k este un răspuns care conține X_i și X_j , atunci el conține și reuniunea $X_{ij} = X_i \cup X_j$, deoarece

$$[\forall x \in X] (\max(\gamma_i(x), \gamma_j(x)) \geq \gamma_i(x)),$$

$$[\forall x \in X] (\max(\gamma_i(x), \gamma_j(x)) \geq \gamma_j(x)),$$

$$[\forall x \in X] (\gamma_k(x) \geq \gamma_i(x)),$$

$$[\forall x \in X] (\gamma_k(x) \geq \gamma_j(x))$$

și deci

$$[\forall x \in X] (\gamma_k(x) \geq \max(\gamma_i(x), \gamma_j(x)) = \gamma_{ij}(x)),$$

ceea ce implică

$$X_{ij} \subset X_k.$$

Definim intersecția a două răspunsuri X_i și X_j cu funcțiile de selecție γ_i și γ_j , ca fiind un răspuns $X_{ij} = X_i \cap X_j$ la care

$$[\forall x \in X] (\gamma_{ij}(x) = \min(\gamma_i(x), \gamma_j(x))),$$

adică cel mai mare răspuns care este conținut în X_i și X_j .

Pe baza relației

$$[\forall x \in X] (1 - \max(\gamma_i(x), \gamma_j(x))) = \min(1 - \gamma_i(x), 1 - \gamma_j(x)),$$

care se poate verifica ca fiind identitate pentru cele două cazuri posibile

$$\gamma_i(x) > \gamma_j(x),$$

$$\gamma_i(x) < \gamma_j(x),$$

se pot verifica identitățile

$$\begin{aligned} X_i \cup X_j &= X_j \cup X_i && \text{comutativitate,} \\ X_i \cup (X_j \cup X_k) &= (X_i \cup X_j) \cup X_k && \text{asociativitate,} \\ X_i \cap (X_j \cap X_k) &= (X_i \cap X_j) \cap (X_i \cap X_k) && \text{distributivitate.} \end{aligned}$$

Această interpretare a răspunsului sistemului ca o mulțime fuzzy permite analiza unitară a strategiilor de selecție, indiferent de faptul că cererea de selecție este liniară (cu descriptori independenți) sau structurată (cu descriptori legați prin relații logice). În paragraful ce urmează se arată cum răspunsul unui sistem cu cerere structurată poate fi interpretat în termenii răspunsului unui sistem cu cerere liniară.

2.9. STRATEGII CU CERERE STRUCTURATĂ

Numim termen, o disjuncție de descriptori

$$q_k = \bigvee d_k(x).$$

O cerere de selecție se zice structurată dacă poate fi reprezentată printr-o conjuncție de termeni

$$q = \bigwedge q_k.$$

O cerere structurată cu i termeni, fiecare termen avînd n_j descriptori este echivalentă cu $z = \prod_{j=1}^i n_j$ cereri liniare. Astfel o cerere de forma

$$q = (d_1 \vee d_2) \wedge (d_3 \vee d_4 \vee d_5) \wedge d_6$$

este echivalentă cu 6 cereri liniare

$$d_1, d_3, d_6,$$

$$d_1, d_4, d_6,$$

$$d_1, d_5, d_6,$$

$$d_2, d_3, d_6,$$

$$d_2, d_4, d_6,$$

$$d_2, d_5, d_6.$$

O cerere de forma

$$q = (d_1 \vee d_2) \wedge (d_3 \vee d_4 \vee (d_5 \wedge d_6) \vee (d_7 \wedge d_8))$$

este echivalentă cu 8 cereri liniare

$$d_1, d_3,$$

$$d_1, d_4,$$

$$d_1, d_5, d_6,$$

$$d_1, d_7, d_8,$$

$$d_2, d_3,$$

$$d_2, d_4,$$

$$d_2, d_5, d_6,$$

$$d_2, d_7, d_8.$$

O cerere structurată echivalentă cu z cereri liniare pretinde z selecții în urma cărora se obțin z răspunsuri parțiale. Pentru z cereri liniare corespund z strategii

$$\sigma_1 = (\gamma, q_1),$$

$$\sigma_2 = (\gamma, q_2),$$

$$\dots$$

$$\sigma_z = (\gamma, q_z).$$

Fiecărei strategii îi corespunde un răspuns parțial X_i . Reuniunea răspunsurilor parțiale definite de cererile de selecție q_1, q_2, \dots, q_z este un răspuns $X_q = X_1 \cup X_2 \cup \dots \cup X_z$ la care

$$[\forall x \in X] (\gamma_q(x) = \max(\gamma_1(x), \gamma_2(x), \dots, \gamma_z(x))).$$

Intersecția răspunsurilor parțiale definite de cererile de selecție q_1, q_2, \dots, q_z este un răspuns $X_c = X_1 \cap X_2 \cap \dots \cap X_z$ la care

$$[\forall x \in X] (\gamma_c(x) = \min(\gamma_1(x), \gamma_2(x), \dots, \gamma_z(x))).$$

2.10. OBSERVAȚII BIBLIOGRAFICE

O excelentă trecere în revistă a tuturor modelelor matematice propuse pentru sistemele de regăsire a informațiilor este făcută de Edmunson [38].

Mooers [102] și Fairthorne [40] s-au ocupat de sisteme de tip (X, D, τ) , unde X este spațiul înregistrărilor, D spațiul descriptorilor și τ o transformare din D în X . Procesul de selecție este definit prin impunerea unor structuri pe spațiul X și a unei transformări τ astfel încât τ aplicată unei cereri de selecție q produce o submulțime a mulțimii X . Spațiul X este descris ca fiind format din toate submulțimile posibile ale mulțimii X , adică mulțimea $P(X)$. Fiindcă mulțimea X este finită, o structură evidentă pentru X se sugerează imediat și anume structura de algebră booleană finită. Spațiului D i se dă adesea o structură mai complicată care depinde de tipul de descriptor folosit. Mooers [102] reprezintă descriptorii ca niște sisteme ordonate parțial cu două elemente. Spațiul D devine atunci produsul cardinal al sistemelor parțial ordonate cu două elemente. Acest spațiu este o latice booleană.

Totuși structurile abstracte impuse pe D și X nu sînt structurile reale induse în D și X de procesul de indexare.

Kasarda [70] arată că într-un sistem de selecție abordarea formală a procesului de selecție nu este foarte utilă și că punctul de plecare în determinarea structurii este chiar procesul de indexare.

Soergel [173] definește o semiordonare a mulțimii D dacă

$$d_1 \leq d_2 \iff [\forall x] (d_1(x) \text{ adevărat} \rightarrow d_2(x) \text{ adevărat})$$

și în acest caz există un element minimal

$$\min \{d \mid d(x) \text{ adevărat}\}.$$

După Mooers el definește două transformări ale mulțimii D în mulțimea $P(X)$ a tuturor submulțimilor lui X

$$T_{\text{ex}}(d) = \{x \mid d(x) \text{ adevărat} \wedge d = \min \{d \mid d(x) \text{ adevărat}\}\},$$

$$T_{\text{in}}(d) = \bigcup \{T_{\text{ex}}(e) \mid e \leq d\} = \{x \mid d(x) \text{ adevărat}\}.$$

Mulțimile T_{ex} nu au nici un element comun deoarece pentru fiecare semiordonare există un singur element minimal. Mulțimile T_{in} , în general, nu sînt disjuncte, deoarece pentru $d_1 \neq d_2$ poate exista

e încât $e \leq d_1$ și în același timp $e \leq d_2$ astfel ca $T_{ex}(e) \subset T_{in}(d_1)$ și $T_{ex}(e) \subset T_{in}(d_2)$.

Fairthorne [40] a introdus două transformări: pseudocomplementul dublu și complementul Browerian dublu. Dacă D_i este o mulțime de descriptori, pseudocomplementul dublu D_i^* al mulțimii D_i este cea mai mică mulțime care conține toate înregistrările indexate de D_i , dar nu numai de D_i . Complementul Browerian dublu $\gg D_i$ este cea mai mare mulțime de înregistrări care conține numai pe D_i , însă nu toate înregistrările ce conțin pe D_i .

Rocchio [146], Salton și Woods [162] au indicat sisteme de tip (X, Q, τ) , unde Q este mulțimea cererilor de selecție, iar τ este o transformare din Q în X sau în $P(X)$.

Goffman [50] a introdus noțiunea de funcție de evaluare $E(A)$ și a definit procesul de selecție ca determinarea unei submulțimi $B \subset P(X)$ astfel ca $E(A)$ să fie maximă pentru $B = A$. El a indicat o funcție de evaluare de forma

$$E(A) = \sum_{x \in A} (ap(x) - b(1 - p(x))),$$

unde $p(x)$ este o probabilitate definită de cererea de selecție. Funcția de evaluare este deci o măsură a premierii minus o măsură a penalizării sistemului, iar a și b sînt constante nenegative ale sistemului.

Noțiunea de strategie de selecție a fost introdusă pentru prima dată de Kent [72], Becker și Hayes [8]. Ei au definit strategia ca forma cererii de selecție. Această accepțiune a noțiunii este întâlnită în majoritatea lucrărilor privind sistemele de regăsire a informațiilor. Salton [162] folosește noțiunea de strategie pentru o metodă de selecție.

O formulare a procesului de selecție folosind reprezentarea vectorială pentru înregistrări și cerere a fost făcută de Salton [158], care a introdus coeficientul de corelație tip cosinus.

Un studiu experimental pentru compararea eficacității coeficienților de corelație folosiți în prezent este făcut de Reitsma-Sagaly [144].

Noțiunile de funcție de selecție și de strategie de selecție, metoda de determinare a eficienței strategiilor de selecție și interpretarea răspunsului unui sistem ca o mulțime fuzzy, au fost introduse de [121], [122], [124]. Echivalența strategiilor de selecție este prezentată pe baza lucrărilor lui Rocchio [146] și Cooper [27].

3 SISTEME CU SELECȚIE PRIN ASOCIERE

În cele ce urmează se consideră asocierea automată a descriptorilor similari, presupunând că similaritatea este determinată de coapariție.

3.1. MATRICEA DE SIMILITUDINE

Fie matricea de fixare

$$F = \begin{bmatrix} v_{11} & v_{21} & \dots & v_{n1} \\ v_{12} & v_{22} & \dots & v_{n2} \\ \cdot & \cdot & \cdot & \cdot \\ v_{1m} & v_{2m} & \dots & v_{nm} \end{bmatrix},$$

unde $v_{ij} \in V$ este valoarea descriptorului $d_i \in D$ pentru înregistrarea $x_j \in X$. Atunci fiecărei coloane i îi corespunde o mulțime finită D_i a înregistrărilor x care conține descriptorul d_i :

$$D_i = \{x | v_{ij} > 0\}.$$

Fie $\mathcal{D} = \{D_1, D_2, \dots, D_n\}$ mulțimea tuturor mulțimilor D_i și $\mathcal{D}^* = \{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_n\}$ mulțimea tuturor submulțimilor mulțimii \mathcal{D} .

O funcție de distanță peste mulțimea $\mathcal{D}^* \times \mathcal{D}^*$ este o aplicație

$$\delta : \mathcal{D}^* \times \mathcal{D}^* \rightarrow R,$$

care fixează fiecărei perechi de mulțimi $(\mathcal{D}_i, \mathcal{D}_j)$ un număr real astfel ca

$$[\forall \mathcal{D}_i] [\forall \mathcal{D}_j] (\delta(\mathcal{D}_i, \mathcal{D}_j) = \delta(\mathcal{D}_j, \mathcal{D}_i)),$$

$$[\forall \mathcal{D}_i] [\forall \mathcal{D}_j] (\mathcal{D}_i = \mathcal{D}_j \Leftrightarrow \delta(\mathcal{D}_i, \mathcal{D}_j) = 0),$$

$$[\forall \mathcal{D}_i] [\forall \mathcal{D}_j] [\forall \mathcal{D}_k] (\delta(\mathcal{D}_i, \mathcal{D}_k) \leq \delta(\mathcal{D}_i, \mathcal{D}_j) + \delta(\mathcal{D}_j, \mathcal{D}_k)).$$

Dacă considerăm mulțimile $\mathcal{D}_i = \{D_i\}$, $\mathcal{D}_j = \{D_j\}$ și notăm

$$\delta(D_i, D_j) = s_{ij}$$

se obține o matrice

$$S = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ s_{21} & s_{22} & \dots & s_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ s_{n1} & s_{n2} & \dots & s_{nn} \end{bmatrix}$$

numită matrice de similitudine ale cărei elemente le vom numi factori de similitudine de tip δ .

În matricea de fixare F fiecare coloană este un vector D_i . Atunci mulțimea $\mathcal{D} = \{D_1, D_2, \dots, D_n\}$ poate fi considerată ca mulțimea vectorilor D_i .

O funcție de apropiere pe mulțimea $\mathcal{D} \times \mathcal{D}$ este o aplicație

$$\alpha: \mathcal{D} \times \mathcal{D} \rightarrow R,$$

care fixează fiecărei perechi ordonate (D_i, D_j) un număr real dacă

$$[\forall D_i] [\forall D_j] (\alpha(D_i, D_j) = \alpha(D_j, D_i)),$$

$$[\forall D_i] [\forall D_j] (D_i = D_j \Leftrightarrow \alpha(D_i, D_j) = 1).$$

În acest caz

$$s_{ij} = \alpha(D_i, D_j)$$

este un factor de similitudine de tip α .

3.2. METODA EXTINDERII CERERII DE SELECȚIE

Matricei S de similitudine îi corespunde o matrice A adiacentă booleană ale cărei elemente 0 sau 1 se determină în modul următor:

$$a_{ij} = 1 \quad \text{dacă} \quad s_{ij} \geq s_0,$$

$$a_{ij} = 0 \quad \text{dacă} \quad s_{ij} < s_0,$$

unde s_0 este o valoare de prag.

Elementele matricii A specifică o corespondență prin care fiecărui descriptor d_i îi corespund descriptorii d_j ,

$$d_i \rightarrow a_{ij} d_j, \quad j = 1, \dots, n, j \neq i.$$

Vom numi descriptori similari descriptorii d_j astfel obținuți. Considerăm o cerere de selecție dată

$$q = \{d_k(q) \mid k = 1, \dots, n\}.$$

Sistemul poate construi automat o nouă cerere de selecție inserind în cererea dată descriptorii similari obținuți din corespondențele

$$d_k(q) \rightarrow a_{kj} d_j, \quad j = 1, \dots, n, j \neq k.$$

Astfel de la strategia $\sigma = (q, \gamma)$ sistemul trece automat la strategia $\sigma_{s_0} = (q_{s_0}, \gamma)$.

3.3. METODA VALORII DE ASOCIERE

Considerăm vectorii

$$x = \{d_k(x) \mid k = 1, \dots, n\},$$

$$q = \{d_k(q) \mid k = 1, \dots, n\}.$$

Pentru fiecare descriptor i nenul din q există

$$v_i^k = \sum_j s_{ij},$$

unde s_{ij} sînt elementele matricii S , iar j sînt indicii descriptorilor nenuli din înregistrarea $x_k \in X$. Pentru fiecare înregistrare x_k există deci

$$v_k = \sum_i v_i^k.$$

unde i sînt indicii descriptorilor nenuli din q .

Valoarea maximă a lui v_k se obține în cazul cînd x_k este un vector cu toate componentele diferite de zero, adică are toți descrip-

torii sistemului. Atunci

$$v_{i,\max} = \sum_{j=1}^n s_{ij},$$

$$v_{\max} = \sum_{i=1}^n v_{i,\max}.$$

Funcția γ_q cu valori

$$\gamma_q(x) = \frac{v_k}{v_{\max}}$$

este o funcție de selecție.

Cu această metodă pentru o cerere q sistemul își determină automat funcția de selecție și deci strategia de selecție.

Într-un sistem cu selecție prin asociere, lucrînd corect, nu mai este necesar să se folosească în cereri descriptori identici cu cei din înregistrări, cu condiția ca în cererea inițială să se folosească descriptorii sistemului. Se presupune însă că relațiile care generează asociația au sens și că sînt generate toate asociațiile.

Practic, aceasta presupune că matricea de fixare, folosită ca să genereze asocieri este descriptivă pentru o colecție mare.

3.4. FACTORI DE SIMILITUDINE

3.4.1. Factori de similitudine tip δ

Fie $\mathcal{D} = (D_1, D_2, \dots, D_n)$, unde D_i este mulțimea înregistrărilor x care conțin descriptorul d_i ,

$$D_i = \{x \mid v_{ij} > 0\},$$

și $\mathcal{D}^* = \{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_n\}$ mulțimea tuturor submulțimilor mulțimii \mathcal{D} . Vom arăta că puterea diferenței simetrice

$$\mathcal{D}_i + \mathcal{D}_j = (\mathcal{D}_i \cup \mathcal{D}_j) - (\mathcal{D}_i \cap \mathcal{D}_j)$$

este o distanță, adică

$$\delta_M(\mathcal{D}_i, \mathcal{D}_j) = \text{card}(\mathcal{D}_i + \mathcal{D}_j).$$

ceea ce înseamnă

$$\text{card} (\mathfrak{D}_i + \mathfrak{D}_j) = 0 \rightarrow \mathfrak{D}_i = \mathfrak{D}_j,$$

$$\text{card} (\mathfrak{D}_i + \mathfrak{D}_j) = \text{card} (\mathfrak{D}_i + \mathfrak{D}_i),$$

$$\text{card} (\mathfrak{D}_i + \mathfrak{D}_k) = \text{card} (\mathfrak{D}_i + \mathfrak{D}_j) + \text{card} (\mathfrak{D}_j + \mathfrak{D}_k).$$

Fiindcă $\delta_M (\mathfrak{D}_i, \mathfrak{D}_j)$ este puterea unei mulțimi, aceasta este fie un număr pozitiv, fie zero dacă mulțimea este vidă. În ultimul caz toate elementele lui \mathfrak{D}_i aparțin lui \mathfrak{D}_j și toate elementele lui \mathfrak{D}_j aparțin lui \mathfrak{D}_i , adică

$$\mathfrak{D}_i = \mathfrak{D}_j$$

și condiția 1 este satisfăcută.

Fiindcă operația diferență simetrică este comutativă, condiția 2 este satisfăcută.

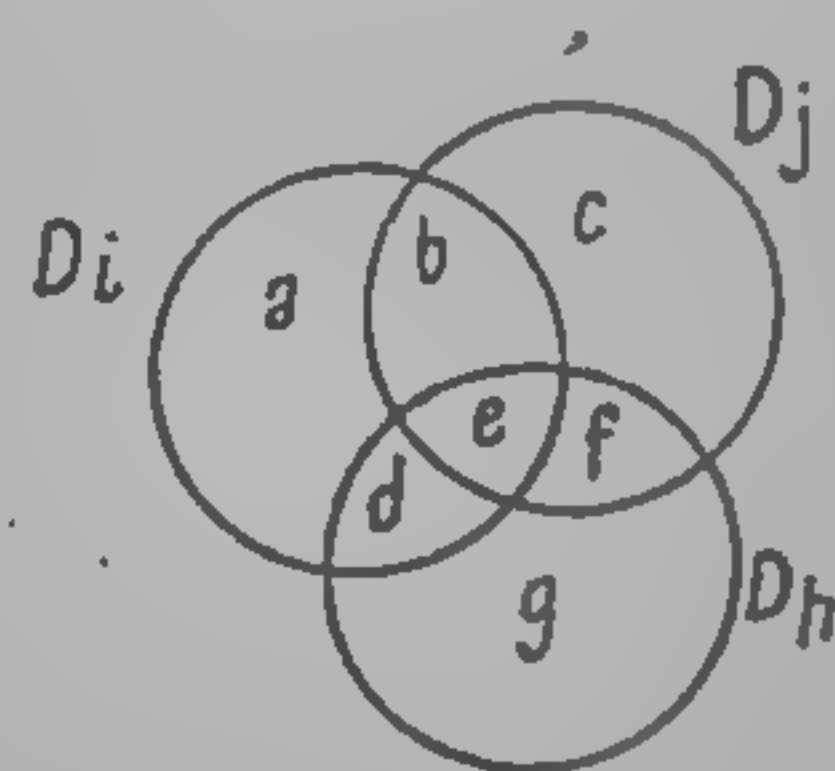
În diagramă fie a, b, c, d, e, f, g diversele regiuni în care $\mathfrak{D}_i, \mathfrak{D}_j, \mathfrak{D}_k$ pot fi împărțite.

Avem

$$\mathfrak{D}_i + \mathfrak{D}_k = \{a, b, f, g\},$$

$$\mathfrak{D}_k + \mathfrak{D}_j = \{b, c, d, g\},$$

$$\mathfrak{D}_i + \mathfrak{D}_j = \{a, d, c, f\}.$$



Deci

$$\delta_M (\mathfrak{D}_i, \mathfrak{D}_k) = \text{card } a + \text{card } b + \text{card } f + \text{card } g,$$

$$\delta_M (\mathfrak{D}_k, \mathfrak{D}_j) = \text{card } b + \text{card } c + \text{card } d + \text{card } g,$$

$$\delta_M (\mathfrak{D}_i, \mathfrak{D}_j) = \text{card } a + \text{card } d + \text{card } c + \text{card } f.$$

Astfel

$$\begin{aligned} \delta_M (\mathfrak{D}_i, \mathfrak{D}_k) + \delta_M (\mathfrak{D}_k, \mathfrak{D}_j) &= \text{card } a + 2 \text{ card } b + \text{card } c + \\ &+ \text{card } d + \text{card } f + 2 \text{ card } g, \end{aligned}$$

$$\delta_M (\mathfrak{D}_i, \mathfrak{D}_k) + \delta_M (\mathfrak{D}_k, \mathfrak{D}_j) - \delta_M (\mathfrak{D}_i, \mathfrak{D}_j) = 2 (\text{card } b + \text{card } g) \geq 0.$$

Deci

$$\delta_M(\mathcal{D}_i, \mathcal{D}_k) + \delta_M(\mathcal{D}_k, \mathcal{D}_j) \geq \delta_M(\mathcal{D}_i, \mathcal{D}_j)$$

și condiția 3 este satisfăcută.

Acum considerăm submulțimile

$$\mathcal{D}_i = \{D_i\},$$

$$\mathcal{D}_j = \{D_j\}$$

și notăm

$$\delta_M(D_i, D_j) = s_{ij}.$$

Se observă că atunci când mulțimile sînt disjuncte

$$D_i \cap D_j = \emptyset,$$

$$D_i + D_j = D_i \cup D_j,$$

$$\max \delta_M(D_i, D_j) = \text{card } D_i + \text{card } D_j.$$

În matricea de fixare F fiecare coloană este un vector D_i .

Funcția

$$\sqrt{\sum_{k=1}^n (d_k(x_i) - d_k(x_j))^2} = \delta_E(D_i, D_j)$$

este o distanță. Primele două condiții sînt vizibil verificate. Pentru cea de-a treia condiție vom folosi inegalitatea lui Cauchy-Bunyakovski :

$$\left[\sum_{k=1}^n a_k b_k \right]^2 \leq \sum_{k=1}^n a_k^2 \sum_{k=1}^n b_k^2.$$

Dacă notăm

$$d_k(x_j) - d_k(x_i) = a_k,$$

$$d_k(x_k) - d_k(x_j) = b_k,$$

atunci

$$d_k(x_k) - d_k(x_i) = a_k + b_k.$$

Cu aceste notații

$$\begin{aligned}
 \delta_E^2(D_i, D_k) &= \sum_{k=1}^n (d_k(x_k) - d_k(x_i))^2 = \sum_{k=1}^n (a_k + b_k)^2 = \\
 &= \sum_{k=1}^n a_k^2 + 2 \sum_{k=1}^n a_k b_k + \sum_{k=1}^n b_k^2 \leq \\
 &\leq \sum_{k=1}^n a_k^2 + 2 \sqrt{\sum_{k=1}^n a_k^2 \sum_{k=1}^n b_k^2} + \sum_{k=1}^n b_k^2 = \left(\sqrt{\sum_{k=1}^n a_k^2} + \sqrt{\sum_{k=1}^n b_k^2} \right)^2 = \\
 &= (\delta_E(D_i, D_j) + \delta_E(D_j, D_k))^2.
 \end{aligned}$$

În cazul vectorilor binari

$$\delta_M(D_i, D_j) = \delta_E(D_i, D_j).$$

3.4.2. Factori de similitudine de tip α

Factorul de similitudine de tip α poate fi definit de oricare din funcțiile de apropiere analizate în paragraful 2.5. De exemplu

$$(s_{ij})_{SM} = \frac{\sum_{k=1}^r d_i(x_k) d_j(x_k)}{\sum_{k=1}^r d_i(x_k)},$$

$$(s_{ij})_C = \frac{\sum_{k=1}^r d_i(x_k) d_j(x_k)}{\left(\sum_{k=1}^r d_i^2(x_k) \sum_{k=1}^r d_j^2(x_k) \right)^{1/2}},$$

$$(s_{ij})_{PRN} = \frac{\sum_{k=1}^r d_i(x_k) d_j(x_k)}{\sum_{k=1}^r d_i^2(x_k) + \sum_{k=1}^r d_j^2(x_k) - \sum_{k=1}^r d_i(x_k) d_j(x_k)},$$

$$(s_{ij})_{MM} = \frac{\sum_{k=1}^r \min(d_i(x_k), d_j(x_k))}{\sum_{k=1}^r \max(d_i(x_k), d_j(x_k))}$$

3.5. FACTORI DE SINONIMIE

3.5.1. Factori de sinonimie bazați pe matricea de similitudine

Considerăm matricea de similitudine

$$S = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ s_{21} & s_{22} & \dots & s_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ s_{n1} & s_{n2} & \dots & s_{nn} \end{bmatrix}$$

și vectorii linie

$$S_{i\cdot} = \{s_{ij} | i = 1, \dots, n\},$$

$$S_{\cdot k} = \{s_{ik} | i = 1, \dots, n\}.$$

Factorul de sinonimie este factorul de similitudine dintre vectorii $S_{i\cdot}$ și $S_{\cdot k}$:

$$t_{jk} = t(S_{i\cdot}, S_{\cdot k}).$$

Folosind din nou funcțiile de apropiere factorul de sinonimie poate avea forma

$$t_{jk} = \frac{\sum_{i=1}^n \min(s_{ij}, s_{ik})}{\min\left(\sum_{i=1}^n s_{ij}, \sum_{i=1}^n s_{ik}\right)},$$

$$t_{jk} = \frac{\sum_{i=1}^n s_{ij} s_{ik}}{\left(\sum_{i=1}^n s_{ij}^2, \sum_{i=1}^n s_{ik}^2\right)^{1/2}}.$$

3.5.2. Factori de sinonimie
bazați pe context

Introducem notațiile :

f numărul total de componente ale unui vector coloană în matricea F ,

f_i numărul de componente nenule ale unui vector coloană în matricea F ,

f_{ij} numărul de componente nenule comune în vectorii coloană i și j în matricea F .

Numim context al descriptorului d_i , mulțimea

$$C_i = \{d_k | f_{ik} \neq 0, k \neq i, k = 1, \dots, n\}.$$

Contextul descriptorului d_j va fi mulțimea

$$C_j = \{d_k | f_{jk} \neq 0, k \neq j, k = 1, \dots, n\}.$$

Numim context al descriptorilor d_i și d_j , mulțimea

$$C_{ij} = \{d_k | f_{ik} \neq 0, f_{jk} \neq 0, k \neq i \neq j, k = 1, \dots, n\}.$$

Fie

$$C_{ijk} = \{d_k | f_{ik} \neq 0, f_{jk} \neq 0, f_{ijk} = 0, k \neq i \neq j, k = 1, \dots, n\}$$

și

$$C_{ijk}^* = \{d_k | f_{ik} \neq 0, f_{jk} \neq 0, f_{ijk} \neq 0, k \neq i \neq j, k = 1, \dots, n\}.$$

Atunci se poate scrie

$$C_i \cap C_j = C_{ij},$$

$$C_{ijk} \cup C_{ijk}^* = C_{ij},$$

$$C_{ijk} \cap C_{ijk}^* = \emptyset,$$

$$\text{card } C_{ijk} + \text{card } C_{ijk}^* = \text{card } C_{ij},$$

$$f_{ij} = 0 \rightarrow f_{ijk} = 0,$$

$$f_{ijk} = 0 \rightarrow C_{ijk}^* = \emptyset,$$

$$C_{ijk}^* = \emptyset \rightarrow C_{ijk} = C_{ij}.$$

Spunem că doi descriptori sînt sinonimi cînd au contexte similare și nu coapar în aceeași înregistrare, adică

$$f_{ij} = 0, \\ C_{ijk} = C_i \cap C_j.$$

Un factor liniar de sinonimie de forma

$$s_{ij} = k \text{ card } C_{ijk} - \text{card } C_i - \text{card } C_j$$

este

$$s_{ij} = \sum_{\substack{k=1 \\ k \neq i \neq j}}^n s_k, \text{ unde } s_k = \begin{cases} \min(f_{ik}, f_{jk}) & \text{dacă } f_{ik} > 0, f_{jk} > 0, \\ -f_{ik} & \text{dacă } f_{ik} > 0, f_{jk} = 0, \\ -f_{jk} & \text{dacă } f_{ik} = 0, f_{jk} > 0, \\ 0 & \text{dacă } f_{ik} = 0, f_{jk} = 0. \end{cases}$$

O măsură care să reflecte și mai mult asocierea semantică este

$$s_{ij} = \sum_{\substack{k=1 \\ k \neq i \neq j}}^n s'_k,$$

unde

$$s'_k = \begin{cases} \text{media} \left(\max \left(0, \left(f_{ik} - \frac{f_i f_k}{f} \right) \right), \max \left(0, \left(f_{jk} - \frac{f_j f_k}{f} \right) \right) \right), & \text{dacă } f_{ik} > 0, f_{jk} > 0, \\ -\max \left(0, \left(f_{ik} - \frac{f_i f_k}{f} \right) \right) & \text{dacă } f_{ik} > 0, f_{jk} = 0, \\ -\max \left(0, \left(f_{jk} - \frac{f_j f_k}{f} \right) \right) & \text{dacă } f_{ik} = 0, f_{jk} > 0, \\ 0 & \text{dacă } f_{ik} = 0, f_{jk} = 0. \end{cases}$$

3.6. REDUCEREA VOCABULARULUI

Considerăm matricea de similitudină S . Acestei matrice îi corespunde o matrice adiacentă A ale cărei elemente 0 sau 1 se

determină în modul următor :

$$a_{ij} = 1 \quad \text{dacă} \quad s_{ij} \geq s_0,$$

$$a_{ij} = 0 \quad \text{dacă} \quad s_{ij} < s_0,$$

unde s_0 este o valoare de prag.

Matricea A este matricea asociată unui graf $G(D, \Gamma)$ format din mulțimea D și o aplicație Γ a mulțimii D în $P(D)$, mulțimea părților lui D , încât

$$d_i \in D,$$

$$\Gamma d_i \in P(D).$$

Astfel

$$a_{ij} = 1 \quad \text{dacă} \quad d_j \in \Gamma d_i,$$

$$a_{ij} = 0 \quad \text{dacă} \quad d_j \notin \Gamma d_i.$$

Fie arcul (i, j) , adică perechea (d_i, d_j) dacă $d_j \in \Gamma d_i$. Mulțimea U a arcelor grafului determină complet aplicația Γ . Din acest motiv

$$G(D, \Gamma) \cong G(D, U).$$

Astfel similitudinea este definită de arcele grafului și

$$(d_i, d_j) \in U \Leftrightarrow a_{ij} \in A.$$

Matricei de similitudine îi corespunde un graf simetric.

O cale în graful $G(D, U)$ este o succesiune de arce astfel încât extremitatea finală a fiecărui arc coincide cu extremitatea inițială a arcului următor.

Numim atingere o relație de semiordonare care este :

reflexivă pentru că fiecare punct al grafului este atins de el însuși printr-o cale de lungime zero ;

tranzitivă pentru că dacă există o cale din d_i în d_j și o cale din d_j în d_k atunci există și o cale din d_i în d_k .

În graful simetric atingerea are și proprietatea de simetrie pentru că dacă d_j este atins din d_i , atunci d_i este atins din d_j . În acest caz atingerea este o relație de echivalență și există o descompunere a mulțimii D în mulțimi disjuncte, adică o partiție.

Considerăm matricea de incidență

$$R = (r_{ij}),$$

unde

$$r_{ij} = 1 \text{ dacă } d_i \text{ este atins din } d_j,$$

$$r_{ii} = 1.$$

Fie \mathcal{R} mulțimea tuturor punctelor în D atinse din d_i , \mathcal{R}_k mulțimea tuturor punctelor atinse din D_i pe o cale a cărei lungime nu depășește k .

Mulțimii \mathcal{R}_k i se asociază o matrice R_k . Dacă A este matricea de asociere a grafului $G(D, U)$, atunci în A^k elementul (i, j) reprezintă numărul de secvențe în $G(D, U)$ de lungime k din d_i la d_j .

Fie A_b^k matricea A^k unde înmulțirea este booleană și I matricea unitate. Atunci R_0 este matricea punctelor atinse numai de ele,

$$R_0 = I;$$

R_1 este matricea punctelor atinse de lungime 1,

$$R_1 = I + A;$$

R_2 este matricea punctelor atinse de lungime 2,

$$R_2 = (I + A + A^2).$$

Deoarece $(I + A)^2 = I + 2A + A^2$ și $(2A)_b = A$

avem

$$(I + A)_b^2 = (I + 2A + A^2)_b = (I + A + A^2)_b,$$

$$R_2 = (I + A)_b^2.$$

Atunci pentru orice întreg pozitiv

$$R_k = (I + A + A^2 + \dots + A^k)_b = (I + A)_b^k.$$

Pentru orice graf cu p puncte $R = R_{p-1}$, deoarece când d_i este atins din d_i trebuie să existe o cale de lungime cel mult $p - 1$ din d_i în d_i .

Un graf care nu este tare conex poate fi caracterizat de matricea de incidență R . Fiind dată o matrice R îi separăm liniile și coloa-

nele. Matricea R se zice descompusă sau redusă dacă poate fi împărțită în submatrice R_{11} , R_{12} , R_{21} , R_{22}

$$R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}$$

astfel ca R_{11} și R_{22} să fie pătrate, iar R_{21} și R_{12} să fie constituite numai din zerouri.

Pentru a se putea obține ușor vîrfurile care aparțin aceleiași submatrice este suficient să se observe că dacă două vîrfuri aparțin aceleiași submatrice, atunci în matricea $(I + A)^k$ liniile care corespund fiecăreia dintre ele sînt identice. Este suficient deci să se numere valorile 1 din fiecare linie și să se grupeze vîrfurile avînd linii identice.

De exemplu fie graful a cărei matrice asociată este

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

atunci matricea R este

$$R = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Rearanjînd coloanele și liniile se poate pune în evidență partiția

$$R = \begin{matrix} & d_1 & d_4 & d_5 & d_{10} & d_2 & d_6 & d_9 & d_3 & d_7 & d_8 \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} & \begin{matrix} d_1 \\ d_4 \\ d_5 \\ d_{10} \\ d_2 \\ d_6 \\ d_9 \\ d_3 \\ d_7 \\ d_8 \end{matrix} \end{matrix}$$

Folosind această tehnică de descompunere, matricea de similitudine poate fi scrisă sub formă evasidiagonală :

$$S = \left[\begin{array}{c|ccc} \boxed{S_1} & 0 & \dots & 0 \\ & 0 & \dots & 0 \\ \hline 0 \dots 0 & \boxed{S_2} & 0 & \dots 0 \\ 0 \dots 0 & & 0 & \dots 0 \\ \hline 0 & \dots & \dots & 0 \\ 0 & \dots & \dots & 0 \end{array} \right] \begin{matrix} \\ \\ \\ \boxed{S_k} \\ \end{matrix}$$

unde S_1, S_2, \dots, S_k sînt submatrice pătratice. Fiecare submulțime S_i reprezintă o familie de descriptori și o vom numi matrice de familie. Această matrice poate fi transformată într-o matrice de tranziție P astfel încît fiecare element p_{ij} este nenegativ și suma tuturor elementelor unui rînd este egală cu unitatea.

Dacă matricea de familie este

$$S_i = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1r} \\ s_{21} & s_{22} & \dots & s_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ s_{r1} & s_{r2} & \dots & s_{rr} \end{bmatrix}$$

atunci matricea de tranziție P , corespunzătoare acesteia va avea elemente

$$p_{ij} = \frac{s_{ij}}{\sum_{j=1}^r s_{ij}}$$

Putem interpreta astfel fiecare element p_{ij} al matricei de tranziție ca probabilitatea de trecere de la descriptorul i la descriptorul j în familia i .

Cu această interpretare probabilistică a similitudinii într-o familie mare, familia are toate caracteristicile unui lanț Markov. De aceea pentru matricea P există un vector unic $w = (w_1, w_2, \dots, w_r)$ astfel încât $wP = w$ și $w_i > 0$ pentru $1 \leq i \leq r$.

Fie S_i o matrice de familie $r \times r$, P o matrice de tranziție $r \times r$ și z un vector linie r -dimensional a cărui componentă i este

$$z_i = \sum_{j=1}^r s_{ij}.$$

Atunci zP este un vector linie r -dimensional a cărei componentă i este $\sum_{j=1}^r s_{ji}$. Deoarece matricea de familie S_i este simetrică, $s_{ji} = s_{ij}$ și $zP = z$. Vectorul w are deci componente

$$w_i = \frac{z_i}{\sum_{j=1}^r z_j}.$$

Acest vector indică faptul că familia este mai des în anumite stări decât în altele. Cu alte cuvinte descriptorii cu valoare mare în w ocupă poziții centrale în familie.

Pe baza acestor considerente apare clar faptul că mulțimea descriptorilor unui sistem de regăsire poate fi controlată în interiorul unei familii.

În concluzie, documentele pot fi caracterizate numai cu descriptorii centrali ai familiei și considerînd toate familiile se poate reduce substanțial lungimea înregistrărilor.

3.7. OBSERVAȚII BIBLIOGRAFICE

Maron [95], [96], [98] a utilizat primul factor de similitudine, iar Stiles [183] și Doyle [36] au indicat extinderea automată a cererii de selecție prin asocierea descriptorilor similari. Stiles a propus factorul

$$(s_{ij})_{st} = \log \frac{(f_{ij} f - f_i f_j - f/2)^2 f}{f_i f_j (f - f_i) (f - f_j)},$$

unde

f este numărul descriptorilor, f_i numărul înregistrărilor cu descriptorul i , f_j numărul înregistrărilor cu descriptorii i și j , folosind distribuția Pearson și valorile marginale ale tablei de contingență 2×2 și corecția Yates pentru eșantioane mici.

Un amplu studiu comparativ al tuturor factorilor de similitudine bazați pe frecvența de coapariție a descriptorilor, experimentați pînă acum este făcut de Jones și Curtice [69], Soergel [173], Becker și Hayes [12].

O nouă clasă de factori de sinonimie este propusă de Lustig [91].

Măsura similitudinii folosind noțiunea de context a fost propusă de Lewis [87].

Edmunson [38] a propus un model topologic al sinonimiei. El consideră că sinonimia este o relație între cuvinte și anume o relație reflexivă, simetrică și tranzitivă.

Dacă se notează $yS_i x$ faptul că cuvîntul y este sinonim în sensul i cu cuvîntul x , atunci se poate defini clasa i de sinonimie a cuvîntului x ca

$$s_i(x) = \{y | y S_i x\}.$$

Această definiție poate fi extinsă la o mulțime arbitrară E de cuvinte

$$s_i(E) = \{y | [\exists x] (x \in E \wedge y S_i x)\}.$$

Vecinătatea $N_i(x)$ a cuvîntului x este definită ca o submulțime a clasei de sinonimie a cuvîntului x care conține de asemenea pe x ,

$$x \in N_i(x) \subseteq s_i(x).$$

Atunci vocabularul are o topologie de vecinătate.

Ideia reducerii vocabularului se datorește lui Hillman (63). Descompunerea matricei de similitudini este efectuată după metoda lui Harary (57).

4 ORGANIZAREA COLECȚIEI

În sistemele de prelucrare a datelor, extragerea informației din memorie se face identificînd acea regiune a memoriei care conține informația, adică identificînd o adresă. Fiind înregistrate cuvintele a_1, a_2, \dots, a_i la adresele b_1, b_2, \dots, b_i , comandînd o adresă b_i se extrage cuvîntul a_i .

Sistemele de regăsire a informației, de care ne ocupăm, trebuie să rezolve problema inversă: furnizarea adreselor în urma comparației dintre o cerere de selecție și informația stocată. Pentru aceasta sînt indicate memoriile cu conținut adresabil la care comparația este făcută fără să fie necesare adrese, adică comenzi de căutare pentru fiecare înregistrare.

Numim colecție totalitatea înregistrărilor existente în memoria sistemului. Numim interogare procesul de identificare a locului unei înregistrări în colecție. Operația de bază într-o interogare este comparația între informația din memorie și informația din cererea de selecție.

Dacă interogarea se face succesiv pentru fiecare înregistrare, atunci colecția se zice cu căutare secvențială. Dacă interogarea se face simultan asupra tuturor înregistrărilor, atunci colecția se zice cu căutare paralelă.

Dacă construcția memoriei este astfel ca la fiecare interogare să se folosească toți descriptorii, memoria se numește tip catalog. Memoriile la care se efectuează căutarea paralelă folosind o selecție arbitrară a descriptorilor se numesc asociative.

4.1. SISTEME CU CĂUTARE SECVENȚIALĂ

Fie r înregistrări înmagazinate în ordine aleatorie și interogate secvențial. Căutăm valoarea medie a numărului de interogări pentru a se obține h înregistrări specificate.

Pentru a se obține o înregistrare din h înregistrări specificate, $h < r$, se parcurg în medie f înregistrări,

$$f = \sum_{g=1}^r g p(g).$$

Considerăm cazul $h = 2$ și enumerăm valorile medii ale numărului de înregistrări interogate pînă la găsirea uneia din cele două înregistrări specificate. Presupunem că am parcurs o înregistrare, adică $g = 1$. Probabilitatea ei este $2/r$ și valoarea medie

$$1p(1) = 1 \frac{2}{r}.$$

Dacă am citit două înregistrări ($g = 2$), probabilitatea primei înregistrări de a nu fi cea dorită este $1 - \frac{2}{r}$, iar probabilitatea celei de-a

doua înregistrări de a fi cea dorită este $\frac{2}{r-1}$. Evenimentele fiind independente

$$2 p(2) = 2 \left(1 - \frac{2}{r}\right) \left(\frac{2}{r-1}\right) = 2 \frac{r-2}{r} \frac{2}{r-1}.$$

În mod similar, dacă am parcurs trei înregistrări,

$$3 p(3) = 3 \left(1 - \frac{2}{r}\right) \left(1 - \frac{2}{r-1}\right) \left(\frac{2}{r-2}\right) = 3 \frac{r-2}{r} \frac{r-3}{r-1} \frac{2}{r-2}.$$

Prin inducție, pentru g înregistrări se obține

$$\begin{aligned} g p(g) &= g \frac{r-2}{r} \frac{r-3}{r-1} \frac{r-4}{r-2} \frac{r-5}{r-3} \cdots \frac{r-g}{r-(g-2)} \frac{2}{r-(g-1)} = \\ &= 2g \frac{(r-2)!(r-4)!}{(r-1-g)!r!} = \frac{2g(r-g)}{r(r-1)} \end{aligned}$$

și

$$f = \sum_{r=1}^r \frac{2gr}{r(r-1)} - \frac{2r^2}{r(r-1)} = \frac{r^2 + r}{r(r-1)} - \frac{r(r+1)(2r+1)}{3r(r-1)} = \frac{r+1}{3},$$

deoarece primul termen al sumei reprezintă suma primelor r numere întregi, iar al doilea termen reprezintă suma pătratelor primelor r numere întregi.

La fel se demonstrează că pentru cazul $h = 3$ trebuie parcurse în medie $\frac{r+1}{4}$ înregistrări.

Astfel numărul mediu de înregistrări parcurse când se caută în r înregistrări o înregistrare oarecare din h înregistrări specificate este

$$\frac{r+1}{h+1}.$$

4.2. METODE DE SCURTARE A TIMPULUI DE INTEROGARE

Am văzut că pentru a selecta o înregistrare din r trebuie interogate în medie $(r+1)/2$ înregistrări. Timpul cerut pentru selecție este proporțional cu produsul dintre numărul mediu de înregistrări interogate și viteza de citire în memorie. Ținând seama de faptul că o înregistrare are circa 1 200 de caractere și că pe un disc citirea a 2 800 de caractere se face în circa 0,05 s, rezultă un timp mediu per înregistrare de 0,02 s. Pentru o colecție de numai 10 000 înregistrări sînt necesare în medie 100 s pentru selecția unei înregistrări specificate. Acest timp este inacceptabil pentru un dialog om — mașină.

Căutarea secvențială a colecțiilor mari devine imposibilă acolo unde se cere un răspuns în timp real.

Pentru eliminarea dezavantajelor colecțiilor cu căutare secvențială s-au propus cîteva soluții. Există astfel trei metode fundamentale de organizare a colecțiilor: metoda dicționarului, metoda listelor și metoda clasificării.

Un dicționar este similar unui fișier. O înregistrare în dicționar conține descriptorul în notație alfabetică și adresele tuturor documentelor pe care le indexează. Căutarea în dicționar înseamnă să se ia descriptorii din cerere și să se obțină o listă cu adresele în care apar acești descriptori. Prin acest procedeu se selecționează numai înregistrările potențial pertinente ignorând restul colecției.

Dicționarul are totuși câteva dezavantaje, unele chiar foarte importante. În spatele fiecărui descriptor nu poate fi memorată fiecare înregistrare, ci numai adresa într-o memorie unde înregistrările sînt înmagazinate integral. Unii descriptori cu sens larg sînt fixați la multe înregistrări, ceea ce conduce la o creștere inacceptabilă a mărimii dicționarului. De asemenea menținerea dicționarului este foarte grea. Cînd se adaugă noi înregistrări, adresele descriptorilor și înregistrărilor complete trebuie să fie întîi extrase și sortate în ordinea descriptorilor. Deci adăugirile nu se pot face simplu prin lipire la capătul dicționarului, ci prin distribuire și intercalare.

Este evident că un dicționar se completează cu o colecție serială și s-ar părea indicat ca sistemul să le mențină pe amîndouă : înregistrările într-o colecție serială și descriptorii principali într-un dicționar care să fie folosit ca index la colecția serială. În acest caz spunem că dicționatul organizează colecția serială conform unei cereri de selecție. Această formă duală de organizare impune însă două restricții. Mai întîi trebuie menținute două depozite. În al doilea rînd cele două depozite trebuie păstrate în fază. Orice schimbare într-un depozit trebuie să fie reflectată precis într-o schimbare în celălalt.

Restricțiile impuse de menținerea a două depozite pot fi eliminate păstrînd avantajele depozitului dual prin folosirea unei tehnici de listă. În acest caz un index indică ultima înregistrare asociată unui descriptor. Această înregistrare conține adresa următoarei înregistrări avînd același descriptor și astfel printr-un lanț de trimiteri sau printr-o listă se ajunge la cea mai veche înregistrare. În felul acesta există atîtea liste cîți descriptori și fiecare înregistrare aparține unui număr de liste, una pentru fiecare descriptor folosit să o reprezinte. Acest procedeu pare economic, deoarece nu pretinde

memorarea unui dicționar. Totuși trebuie prevăzut spațiu în memorie pentru adresele lanțurilor. Deoarece numărul înregistrărilor depășește numărul descriptorilor, practic nu se salvează spațiu.

Fie că este vorba de o colecție dicționar, fie că este vorba de o colecție organizată în liste, căutarea pornește de la o adresă: adresa listei. Această adresă este tocmai descriptorul din cererea de selecție.

De obicei descriptorul este înregistrat alfanumeric, iar mulțimea descriptorilor sistemului este ordonată. Numim arhivă o mulțime de descriptori ordonată. Pentru a găsi o adresă, căutarea începe la jumătatea arhivei marcând înregistrarea găsită acolo. Dacă valoarea găsită este mai mare sau mai mică decât cea căutată, căutarea se va muta respectiv în jos sau în sus cu o pătrime de arhivă, apoi cu o optime etc., pînă cînd s-a găsit descriptorul dorit sau pînă cînd a rămas un segment atît de mic, încît devine economică căutarea secvențială. După b partiții cînd a rămas de examinat numai un singur descriptor, arhiva de n descriptori a fost tăiată în jumătate de b ori, deci

$$\frac{n}{2^b} = 1$$

și numărul de interogări este

$$b = \log_2 n.$$

Avantajul ordonării este evident, deoarece fără ordonare o interogare secvențială ar pretinde $b = (n + 1)/2$ interogări în medie.

Pentru a ilustra metoda de stabilire a unei adrese, considerăm o arhivă cu n descriptori memorată pe discuri cu o capacitate de k înregistrări pe pistă și k piste pe disc. Memoria va avea deci n/k^2 discuri. Presupunem că arhiva este aranjată pe un cîmp alfabetic cu lungime de 10 caractere, în care fiecare caracter are aceeași probabilitate și că valorile cîmpului sînt unice (adică doi descriptori nu au aceeași valoare a cîmpului).

Căutarea se poate face direct asupra arhivei pe care o notăm A_0 sau prin intermediul unor indexuri numite arhive de indexare.

Arhiva de indexare A_1 are n înregistrări, una pentru fiecare înregistrată a arhivei principale A_0 . Fiecare înregistrare în A_1 reprezintă o adresă a câmpului de 6 cifre binare (două pentru numărul discului, două pentru numărul pistei și două pentru înregistrarea în pistă) a înregistrării în arhiva principală A_0 , însă adresa nu va fi înmagazinată explicit. Când am găsit înregistrarea dorită în A_1 , vom folosi poziția ei sau numărul de secvență în arhivă pentru a găsi adresa în arhiva principală, salvând memoria necesară pentru a înmagazina adresele înregistrate în A_0 .

Pentru a căuta în indexul A_1 este creat un alt index A_2 care dă adresa exactă a primei înregistrări în A_1 pentru fiecare trigramă prezentă în arhivă.

Indexul A_2 conține 26^3 înregistrări, de trei caractere plus o adresă de 4 sau 6 caractere. Lungimea adresei, în acest caz, este o funcție de cât spațiu de memorare este fixat și dacă, de exemplu, o nouă trigramă începe totdeauna într-o pistă nouă.

Un al treilea index A_3 , dă adresa în A_2 a primei înregistrări care să înceapă cu o nouă literă inițială.

Acest index constă din numai 26 de adrese de 4 sau 6 caractere și va fi căutat înscriind litera inițială a unui descriptor într-un index de calculator sau registru de modificare a adresei și regăsind direct adresa corespunzătoare în A_2 .

Funcționarea acestui grup de indexuri este ilustrată în figura 1. Aici un descriptor dintr-o cerere de selecție constă dintr-o valoare de câmp CQTDACRSBJ. În prima etapă, litera inițială este folosită pentru a localiza a treia înregistrare în A_3 . Această înregistrare conține adresa primei intrări în A_2 , începând cu litera C. Pornind de la această adresă se face o căutare binară în domeniul C, localizînd în final o înregistrare care conține primele trei litere ale descriptorului, CQT și o adresă în A_1 la începutul tuturor înregistrărilor CQT. Această adresă este folosită pentru a localiza prima intrare în A_1 , începînd cu CQT. În final se face din

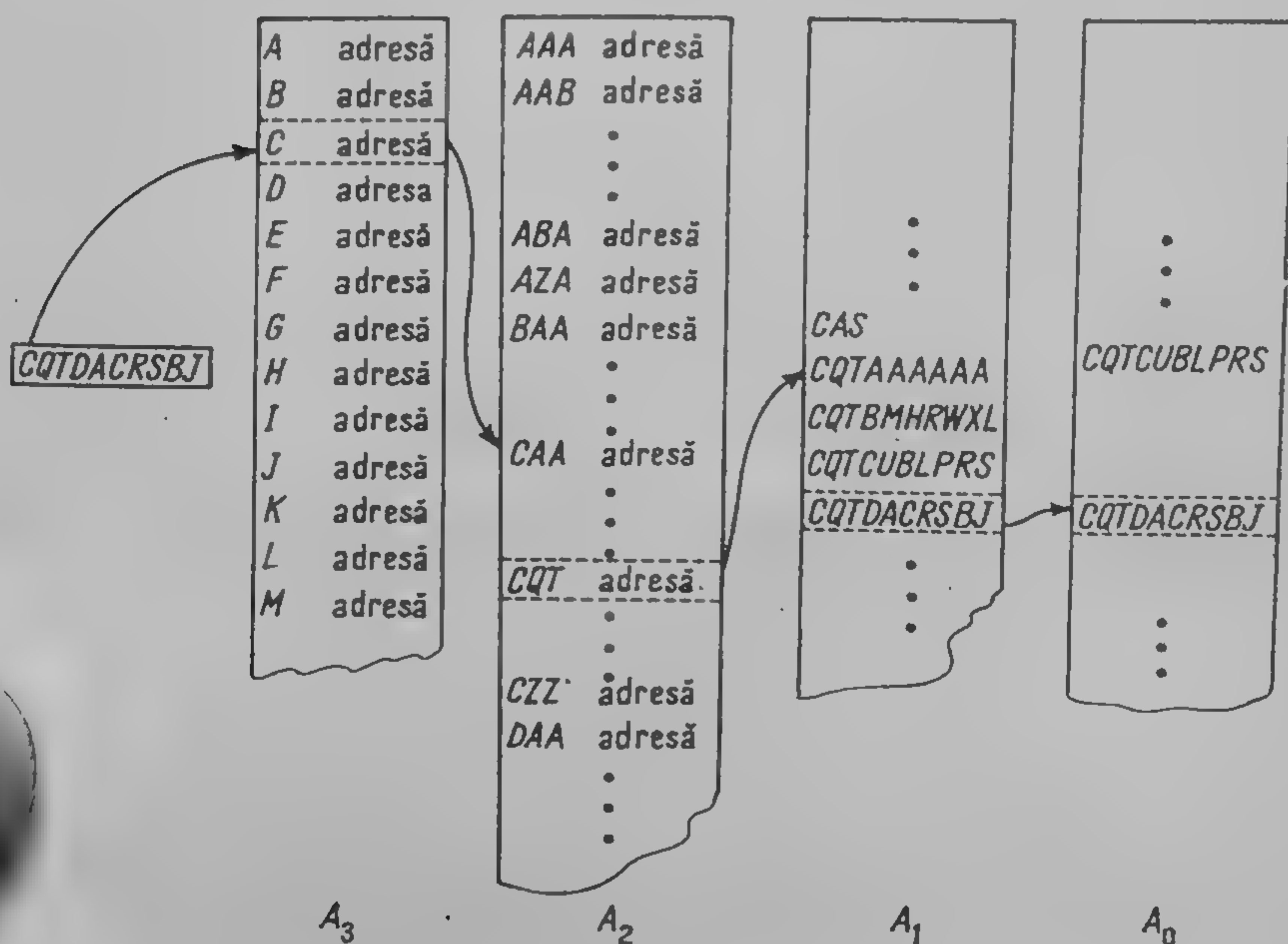


Fig. 1

nou o căutare binară pentru a găsi toată valoarea căutată *CQTDACRSBJ*.

În această etapă nu e nevoie să se găsească explicit o adresă pentru că dacă se știe locul în A_1 se poate folosi această informație pentru a găsi poziția în A_0 .

Numărul exact de interogări efectuate este necunoscut depinzând de numărul de înregistrări în A_2 care încep cu C și de numărul de înregistrări în A_1 începând cu CQT.

În felul acesta există o interogare în A_3 , cel mult $\log_2 26^3$ interogări în A_2 (se știe locul literei inițiale și se caută prin maximum 26×26 litere secundare și terțiare cu metoda binară) și $\log_2 n/26^3$ interogări în A_1 , cele n înregistrări ale lui A_1 fiind împărțite de A_2 în 26^3 subgrupe cu loc cunoscut. În A_0 se face numai o căutare. În total se efectuează $\log_2 n - \log_2 26$ interogări.

Metoda ilustrată mai sus presupune că sînt cunoscute toate valorile descriptorilor.

Este posibilă și o metodă când se cunoaște numai valoarea descriptorilor în prima înregistrare a fiecărei piste. În acest caz se folosește numai arhiva A_1 împreună cu A_0 , ambele avînd lungimea n . În A_1 se efectuează $\log_2 n/k$ interogări, iar în A_0 $\log_2 k$ interogări, deci în total $\log_2 n$ interogări. O altă metodă presupune că se cunoaște valoarea descriptorilor în prima listă a primei piste a fiecărui disc. Și în acest caz se folosesc arhivele A_1 și A_0 , A_1 cu n/k elemente, iar A_0 cu n elemente. În arhiva A_1 se efectuează $\log_2 n/k^2$ interogări, iar în arhiva A_0 , $\log_2 k^2$ interogări, deci în total $\log_2 n$ interogări.

Se observă că se ajunge aproape la același rezultat prin toate cele trei metode care folosesc un index și căutarea binară.

4.2.1. Organizarea în listă

Scopul acestui paragraf este să prezinte unele efecte ale caracteristicilor colecției organizată în liste asupra timpului de selecție.

În figura 2 se arată organizarea unei memorii în liste de înregistrări legate.

Fiecare listă corespunde unui descriptor și toate înregistrările caracterizate de acel descriptor corespund unui nod pe listă. Dacă un document este caracterizat de cîțiva descriptori, înregistrarea este reprezentată de un nod care este intersecția listelor corespunzînd descriptorilor respectivi. În figura 3 se ilustrează schematic structura unei înregistrări și anume înregistrarea 2 din figura 2.

Această înregistrare corespunde unui document caracterizat de descriptorii d_1 , d_2 și d_3 . Deci nodul corespunzător acestui document este pe trei liste și anume listele d_1 , d_2 și d_3 . În acest caz nodul marchează sfîrșitul listei d_1 și precede imediat înregistrarea 7 pe lista d_2 sau înregistrarea 6 pe lista d_3 .

Notăm cu t , timpul de acces al memoriei, adică timpul necesar pentru a transfera un bloc de cuvinte din memoria externă în memoria internă a calculatorului, cu n numărul descriptorilor sistemului, adică numărul cardinal al mulțimii D și cu $f(j)$ numărul înregistrărilor care conțin descriptorul d_j .

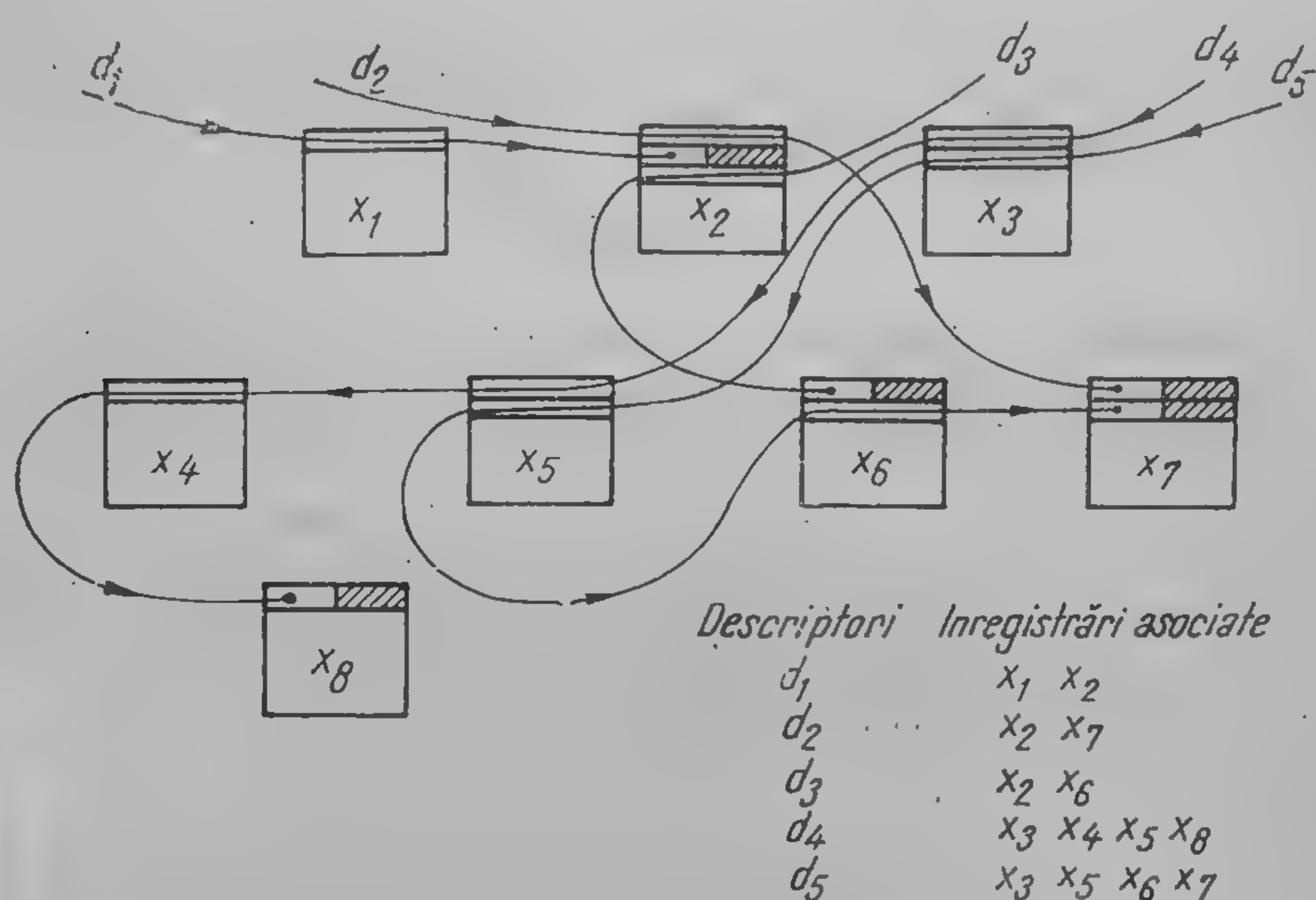


Fig. 2

În structura listei d , există $f(j)$ noduri. Spunem deci că lista d , are lungime $f(j)$.

Timpul cerut pentru a regăsi toate înregistrările pe o listă de lungime $f(j)$ este $f(j)t_i$.

În cele ce urmează se presupune că fiecare înregistrare ocupă numai o locație. În caz contrar, pentru noduri care reprezintă înregistrări lungi ar fi necesar mai mult decât un acces la memorie. O analiză mai completă ar trebui să țină seama de efectul mărimii locației.

Vom căuta acum o expresie pentru timpul necesar regăsirii tuturor înregistrărilor corespunzătoare unui descriptor dat:

Fie s numărul total de apariții al tuturor descriptorilor sistemului. Atunci

$$s = \sum_{j=1}^n f(j).$$

d_1	Adresa lui x_7
d_2	Simbol de sfârșitul listei
d_3	Adresa înregistrării 6
Date	

Fig. 3

Presupunem acum o cerere de selecție formulată cu un singur descriptor și fie $p(j)$ probabilitatea ca cererea să conțină descriptorul j . Atunci timpul mediu de selecție per descriptor este

$$t_r = t_s \sum_{j=1}^n f(j) p(j).$$

Funcțiile $f(j)$ și $p(j)$ sînt necunoscute sau dificil de determinat, depinzînd de natura informației în colecție, de metoda de indexare, de faptul că au fost suprimați descriptorii cu frecvență foarte mare sau foarte mică etc.

În analiza ce urmează vor fi făcute trei presupuneri asupra funcțiilor $f(j)$ și $p(j)$, deoarece cele mai multe combinații ale acestor funcții întîlnite în practică se găsesc între aceste limite.

În primul caz se presupune că funcțiile $f(j)$ și $p(j)$ sînt distribuite uniform și sînt constante pentru toți j . Atunci

$$f(j) = \frac{s}{n},$$

$$p(j) = \frac{1}{n},$$

astfel că

$$\frac{t_r}{t_s} = \frac{s}{n}.$$

Raportul t_r/t_s este timpul de răspuns normalizat.

Raportul s/n va depinde de numărul total de documente indexate și de tehnica de indexare folosită. Variația acestui raport pentru o colecție tipică este dată în figura 4.

Fie j rangul descriptorului, descriptorul cel mai frecvent folosit fiind considerat de rang 1 etc. În cazul cînd doi sau mai mulți descriptori au același rang, li se fixează arbitrar numere consecutive.

Reprezentarea funcției de frecvență $f(j)$ pentru o colecție tipică este arătată în figura 5.

Pentru comunicațiile scrise au fost propuse diverse expresii pentru a reprezenta analitic relația rang-probabilitate. Legile lui Zipf și Mandelbrot sînt cele mai cunoscute. Așa cum a fost formulată

inițial legea lui Zipf ea se aplică la texte scrise în limba engleză și este dată de expresia

$$p(j) = 0,1 j^{-1}.$$

În acest caz suma probabilităților tuturor cuvintelor este 1 numai

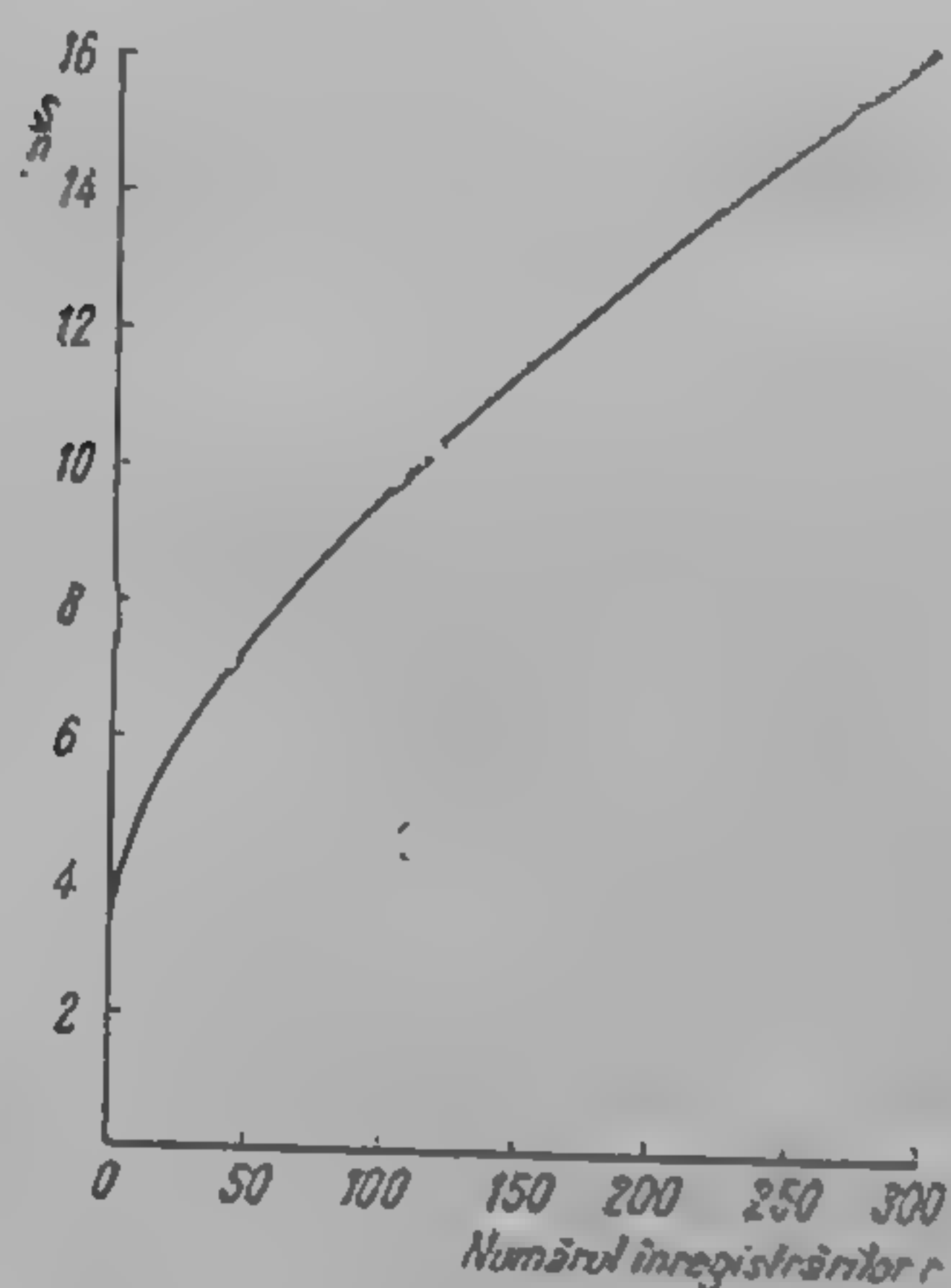


Fig. 4

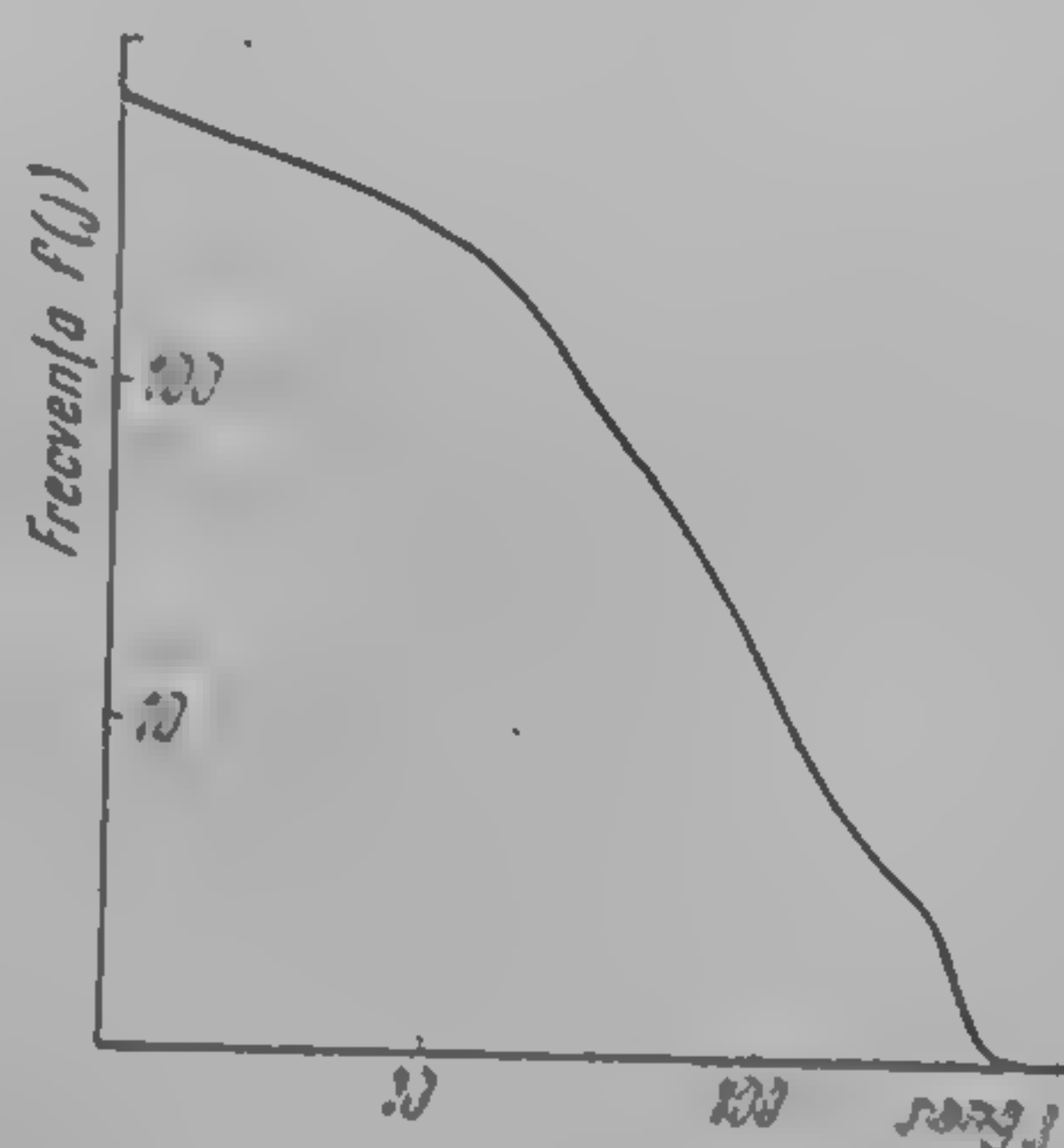


Fig. 5

cînd vocabularul are 8 727 cuvinte. Deoarece în cazul sistemelor de regăsire numărul descriptorilor este cunoscut, o lege modificată se obține ușor, considerînd forma generală a legii

$$p(j) = k j^{-1}$$

și calculînd coeficientul k cu constrîngerile

$$\sum_{j=1}^n f(j) = s,$$

$$\sum_{j=1}^n p(j) = 1.$$

Pentru n mare aproximarea

$$\sum_{j=1}^n \frac{1}{j} = \ln n + c,$$

unde c este constanta lui Euler (0,5772), dă o eroare de 2% pentru $n = 10$ și 0,06% pentru $n = 100$.

A doua presupunere pe care o facem este că funcția $f(j)$ urmează legea lui Zipf, adică

$$f(j) = \frac{s}{j(\ln n + c)}.$$

Fiindcă suma probabilităților trebuie să fie 1 urmează că

$$p(j) = \frac{1}{j(\ln n + c)}.$$

În figura 6 se arată o reprezentare normalizată a funcției f pentru o colecție tipică.

Dacă $f(j)$ urmează legea Zipf și $p(j)$ este uniform

$$t_r = t_s \sum_{j=1}^n f(j) p(j) = t_s \sum_{j=1}^n \left(\frac{s}{\ln n + c} \cdot \frac{1}{j} \cdot \frac{1}{n} \right),$$

adică

$$\frac{t_r}{t_s} = \frac{s}{n}.$$

Se observă că timpul mediu de răspuns este același cu cel din cazul precedent.

În cazul celei de a treia presupuneri ambele funcții urmează legea Zipf și

$$t_r = \frac{st_s}{(\ln n + c)^2} \sum_{j=1}^n \frac{1}{j^2}.$$

Fiindcă seria $1 + 1/4 + 1/9 + 1/16 + \dots$ converge rapid și n este considerat mare, valoarea sumei poate fi luată pentru o serie infinită, adică

$$\frac{t_r}{t_s} = \frac{\pi^2 s}{6 (\ln n + c)^2}.$$

Eroarea produsă prin înlocuirea sumei $\sum_{j=1}^n 1/j^2$ prin $\pi^2/6$ este de 2% pentru $n = 30$ și de circa 0,6% pentru $n = 100$.

Raportul celor două valori t_1/t_2 obținute când $p(j)$ este uniform și când $p(j)$ urmează legea lui Zipf este

$$W = \frac{n \pi^2}{6(\ln n + c)^2}$$

arătat în figura 7.

Se vede că comportarea Zipf a funcției $p(j)$ poate conduce la

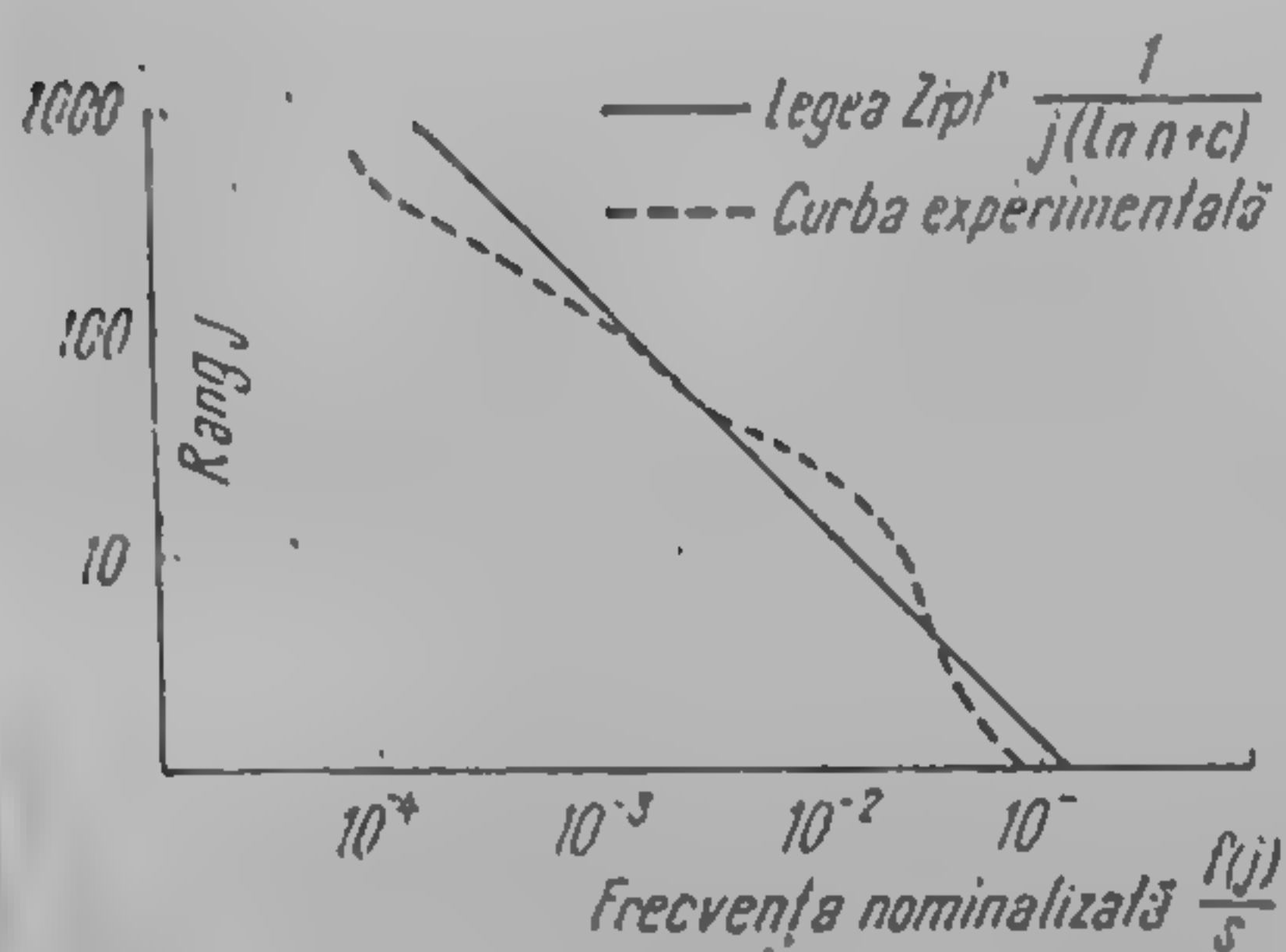


Fig. 6

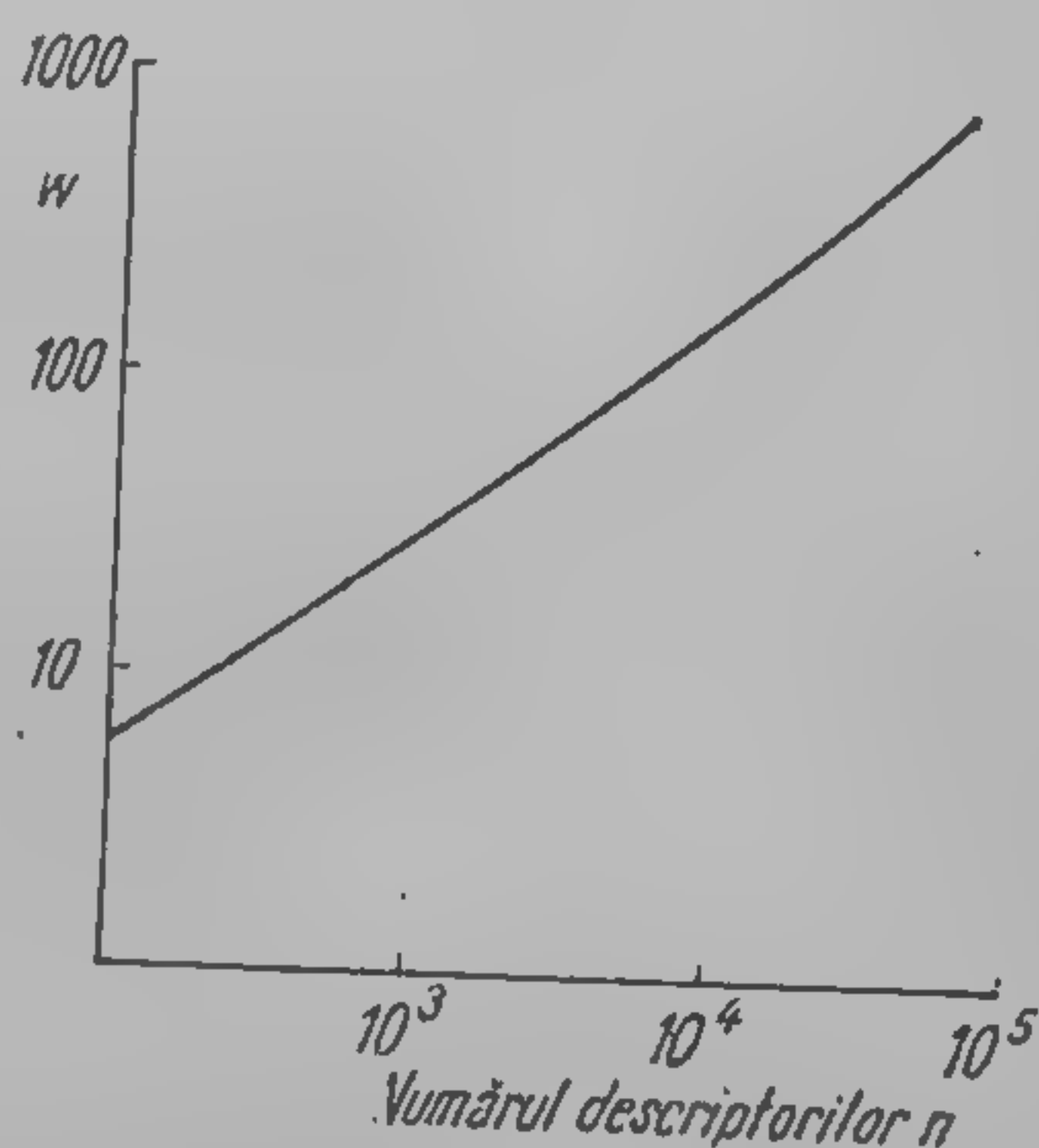


Fig. 7

creșterea timpului de răspuns. Într-o situație reală funcția $p(j)$ va avea o comportare cuprinsă între limitele cazului uniform și cazului Zipf. Deci timpul de răspuns va fi cuprins între valorile s/n și ws/n .

Într-o primă aproximație, cerințele de memorie ale unei liste depășesc pe cele ale unei colecții secvențiale numai datorită cerințelor pentru legături. Dacă includem simbolurile de terminare ale listei atunci vor exista s astfel de legături.

4.2.2. Organizarea în dicționar

În cazul în care colecția este organizată în dicționar, fiecare descriptor reprezintă adresa unei liste care de data aceasta este formată din adresele tuturor înregistrărilor ce reprezintă documente caracterizate de descriptorul respectiv.

În acest paragraf pentru acest mod de organizare se va analiza o relație între timpul de regăsire și parametrii de proiectare ai colecției.

De asemenea va fi analizată o măsură a eficienței de folosire a memoriei, fiindcă în acest caz este posibil să se proiecteze un sistem în care spațiul de memorare alocat nu este folosit integral.

Fie c numărul de adrese care pot fi înmagazinate într-o locație a memoriei, adică capacitatea locației.

O listă asociată cu un anumit descriptor poate să conțină mai mult decât c adrese și în acest caz lista trebuie extinsă peste câteva locații. Invers, dacă există câteva liste scurte, ele pot fi „împachetate” într-o singură locație. Dacă această împachetare nu este făcută porțiunea de locație nefolosită constituie un spațiu pierdut.

O adresă poate fi formată din m caractere și cerința minimă de spațiu de memorare este produsul dintre numărul total de adrese și spațiul de memorare al unei adrese. Notăm cu sm acest produs.

Fie v , memoria totală cerută pentru înmagazinarea dicționarului incluzând memoria pierdută. Atunci o măsură a utilizării memoriei, care este egală cu zero când nu apar pierderi și egală cu unitatea când se alocă dublul spațiului față de cel strict necesar este

$$h = \frac{v - sm}{sm}.$$

Acest factor reprezintă măsura care indică reduceri potențiale de cerințe de capacitate de înmagazinare, reduceri ce se pot obține prin „împachetare”.

În această organizare, parametrii cei mai importanți sînt numărul n de descriptori și raportul s/c dintre numărul tuturor aparițiilor descriptorilor și capacitatea locației.

Cu ajutorul acestor parametrii se pot obține expresii cu care pentru o colecție dată să se determine aplicabilitatea organizării în dicționar. Aceste expresii pot ajuta și la alegerea unei anumite mărimi a locației cu toate că această mărime este determinată de echipamentul folosit. Factorul de utilizare al memoriei h poate fi folosit la luarea de decizii privind împachetarea.

În figura 8 se ilustrează schematic un exemplu de fixare în memorie fără împachetare.

În această figură există n descriptori și deci n liste inversate. Cea mai lungă listă are elemente care cer mai mult decât $4c$ însă mai puțin decât $5c$ unități de memorie, astfel că pentru această listă sînt fixate 5 locații. Rangul celei mai mici liste inversate care cere mai mult decât o locație este notat cu n^* . În anumite cazuri considerate în cele ce urmează n^* va fi legat direct de parametrii

fundamentali n și s/c , fiind util în dezvoltarea expresiilor ce leagă acești parametri de performanțele sistemului.

În figura 8 spațiul de memorare pierdut este reprezentat de intervalul dintre liniile pline și cele punctate. Se observă că acest spațiu poate fi redus micșorând capacitatea c . Totuși, procedînd astfel, pentru a regăsi o singură listă care să acopere cîteva locații va fi nevoie de mai multe accese t_s .

Notînd cu $[y]$ cel mai mic întreg mai mare sau egal cu y și cu $\lceil y \rceil$ cel mai mare întreg mai mic sau egal cu y , din figura 8 se poate vedea că memoria totală v_s fixată pentru listele inversate cînd nu este făcută nici o împachetare este

$$v_s = mc \sum_{j=1}^n \left\lceil \frac{f(j)}{c} \right\rceil$$

Cel mai mic timp de răspuns pentru o memorie este egal cu un timp de acces t_s , iar valoarea minimă a factorului de utilizare h apare cînd nu se pierde nici un spațiu de înmagazinare. Aceste două restricții pot fi formulate astfel

$$\frac{t_r}{t_s} \geq 1,$$

$$h \geq 0.$$

Cu relațiile de mai sus h poate fi scris astfel :

$$h = \frac{c}{s} \sum_{j=1}^n \left\lceil \frac{f(j)}{c} \right\rceil - 1.$$

Cum expresia din sumă nu poate fi mai mică decît unitatea, suma nu poate fi mai mică decît n .

Această constatare permite să se impună o a doua restricție lui h :

$$h \geq n \frac{c}{s} - 1.$$

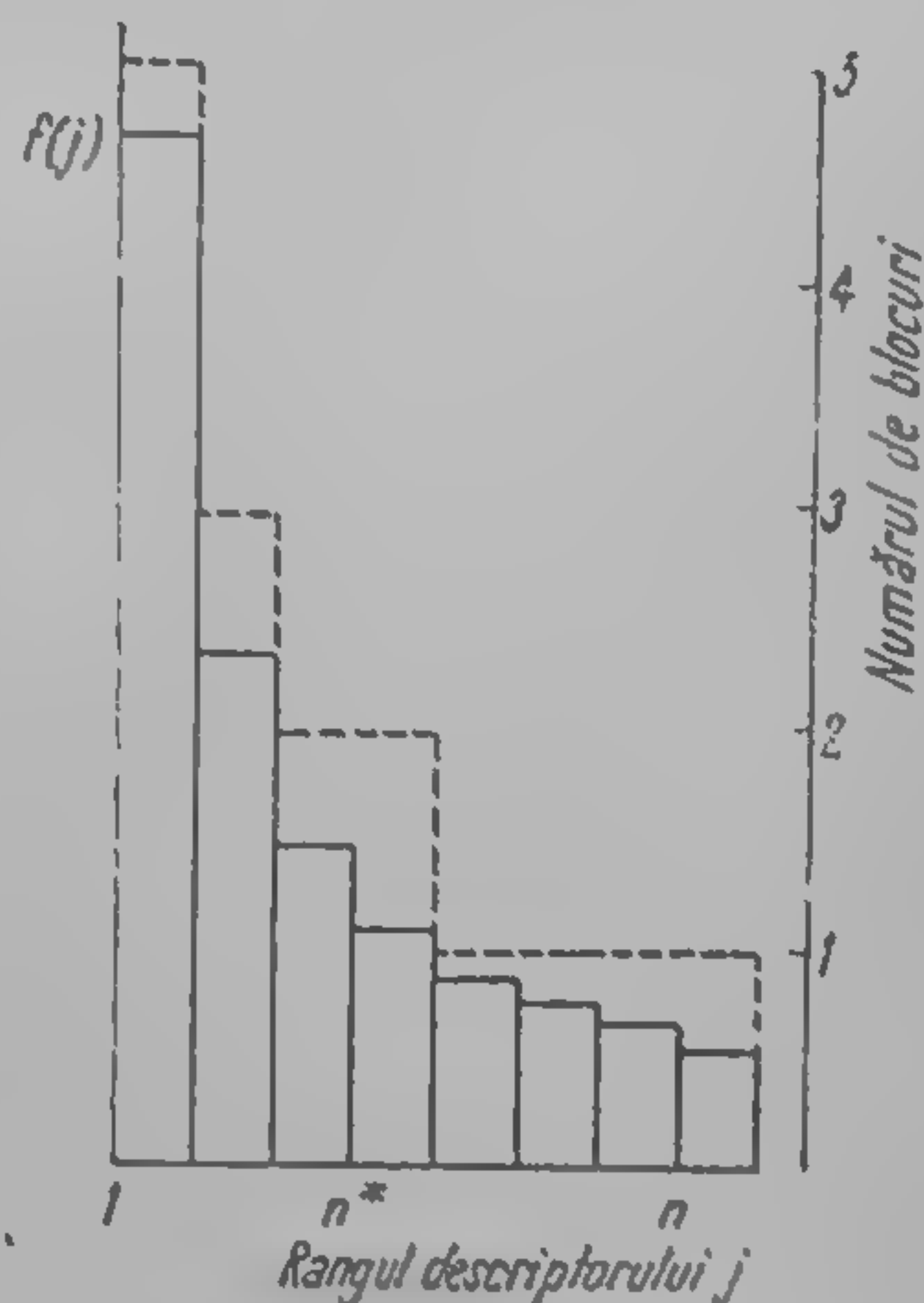


Fig. 8

Pentru a scoate o listă inversată din memorie se cere un acces pentru fiecare locație astfel că timpul mediu de răspuns este dat de expresia

$$t_r = t_s \sum_{j=1}^n \left[\frac{f(j)}{c} \right] p(j).$$

Presupunînd că $f(j)$ și $p(j)$ sînt uniforme

$$f(j) = \frac{s}{n},$$

$$p(j) = \frac{1}{n},$$

și

$$\frac{t_r}{t_s} = \sum_{j=1}^n \left[\frac{s}{c} \frac{1}{n} \right] \frac{1}{n}.$$

În acest caz

$$\frac{1}{n} \frac{s}{c} \leq \frac{t_r}{t_s} < \frac{1}{n} \frac{s}{c} + 1,$$

$$0 \leq h < n \frac{c}{s}.$$

Datorită restricțiilor $h \geq 0$, $h \geq n \frac{c}{s} - 1$, relația de mai sus devine

$$n \frac{c}{s} - 1 \leq h < n \frac{c}{s}.$$

În felul acesta au fost stabilite limitele factorului h în cazul uniform.

În figura 9 este prezentată variația raportului t_r/t_s în funcție de n pentru diverse valori ale raportului s/c , iar în figura 10 este prezentat h .

Liniile exterioare notate „uniform” corespund cazului discutat mai sus. Totdeauna cînd $f(j)$ este uniform, h poate fi minimizat selectînd o mărime a locației astfel ca $c = s/n$. Fiindcă alegerea lui c este limitată de echipamentul fizic, nu totdeauna poate fi posibil să se găsească o valoare optimă într-o situație specifică.

Presupunem acum că $p(j)$ este uniform și $f(j)$ urmează legea lui Zipf. Atunci

$$f(j) = \frac{s}{j(\ln n + k)}$$

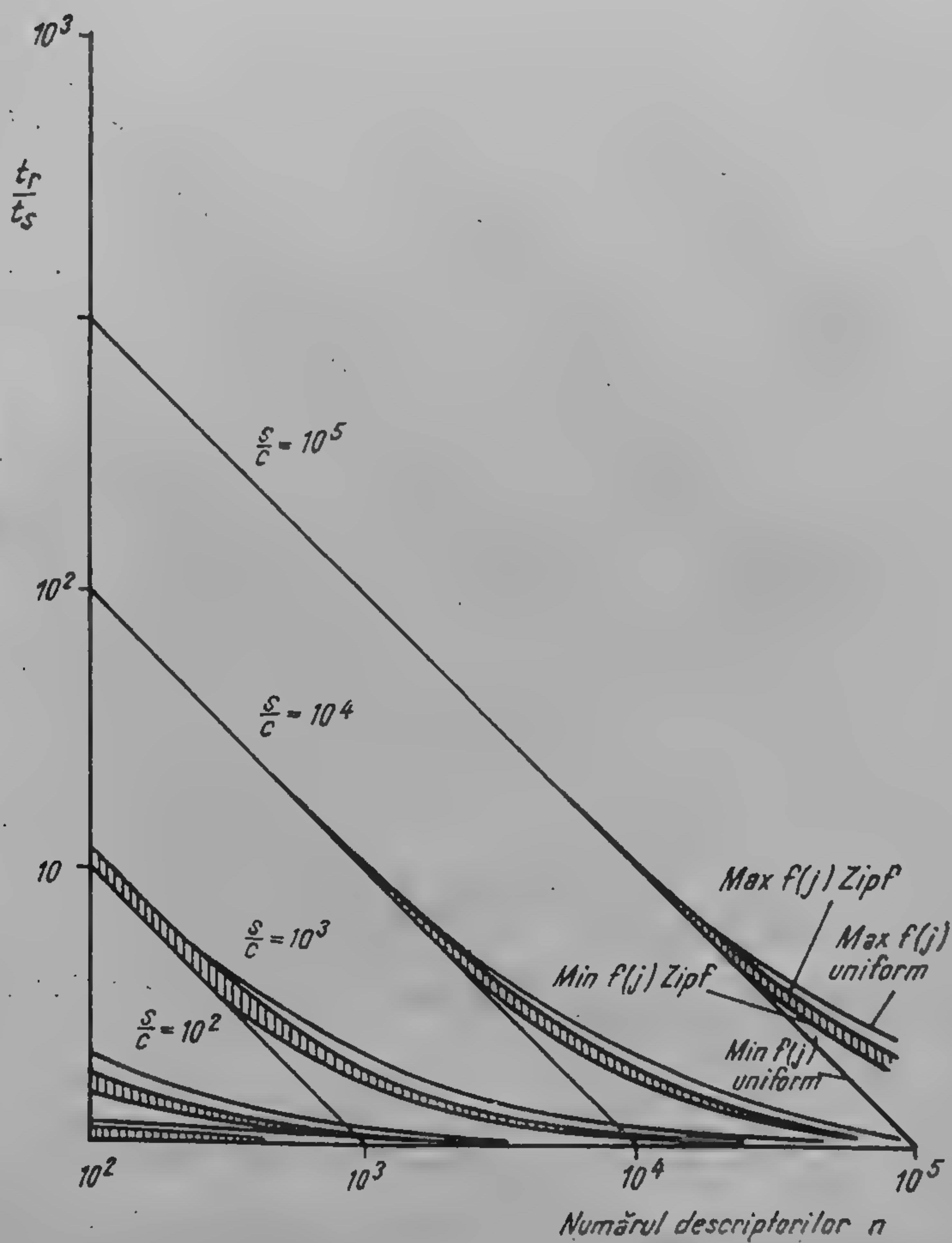


Fig. 9

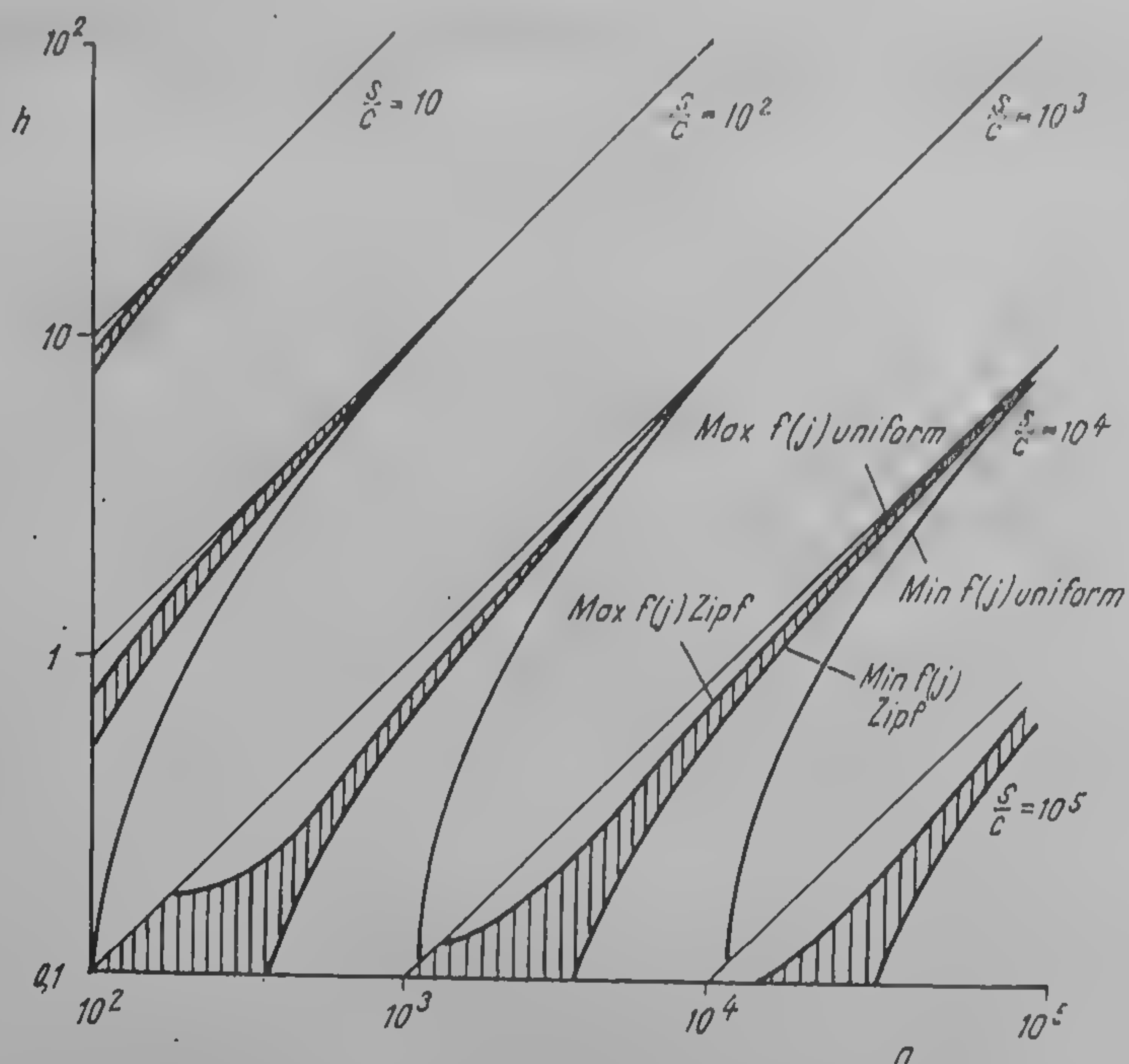


Fig. 10

și

$$\frac{t_r}{t_s} = \frac{1}{n} \sum_{j=1}^n \left[\frac{s}{c} \frac{1}{j(\ln n + k)} \right].$$

Ecuția conduce la aceleași limite pentru t_r/t_s ca în cazul precedent când $f(j)$ era uniform, adică

$$\frac{1}{n} \frac{s}{c} \leq \frac{t_r}{t_s} \leq \frac{1}{n} \frac{s}{c} + 1.$$

În cazul pe care îl analizăm pentru o anumită bandă a parametrilor pot fi obținute limite mai fine. Mai înainte a fost definit și ilustrat

în figura 8, n^* . Din expresia

$$f(j) = \frac{s}{j(\ln n + k)}$$

se obține

$$n^* = \left\lfloor \frac{\frac{s}{c}}{\ln n + k} \right\rfloor.$$

În toate expresiile în care apare n^* este presupus că

$$1 \leq n^* \leq n.$$

În ecuația

$$t_r = t_s \sum_{j=1}^n \left\lfloor \frac{f(j)}{c} \right\rfloor p(j)$$

pentru $j > n^*$, $\left\lfloor \frac{f(j)}{c} \right\rfloor = 1$ și atunci

$$t_r = \frac{t_s}{n} \left(\sum_{j=1}^n \left\lfloor \frac{s}{c} \frac{1}{j(\ln n + k)} \right\rfloor + n - n^* \right).$$

Această expresie permite stabilirea unor limite când n îndeplinește condițiile de mai sus :

$$\frac{1}{n} \left(\frac{s}{c} \frac{1}{\ln n + k} \sum_{j=1}^n \frac{1}{j} - n^* \right) + 1 \leq \frac{t_r}{t_s} < \frac{1}{n} \left(\frac{s}{c} \frac{1}{\ln n + k} \sum_{j=1}^n \frac{1}{j} \right) + 1.$$

În general suma $\sum_{j=1}^n 1/j$ nu poate fi exprimată într-o formă aproxi-

mativă închisă deoarece despre n^* se știe numai că se găsește între 1 și n . În evaluarea numerică a expresiilor pentru limitele lui h sau t_r/t_s , suma este aproximată oricând este posibil. În figura 9 sînt arătate valorile selectate ale parametrilor.

Calculul factorului de utilizare a memoriei h prin substituția formei Zipf a funcției $f(j)$ conduce la aceeași ecuație ca și în cazul

cînd $f(j)$ a fost uniformă. Cînd pentru $j > n^*$ se aplică $\left[\frac{f(j)}{c} \right] = 1$, rezultă

$$h = \frac{c}{s} \sum_{j=1}^n \frac{s}{c} \frac{1}{\ln n + k} \frac{1}{j} - 1.$$

Limitele obținute din expresia de mai sus sînt ilustrate în figura 10. Ele sînt

$$\frac{c}{s} (n - n^*) + \frac{1}{\ln n + k} \sum_{j=1}^{n^*} \frac{1}{j} \leq h \leq \frac{c}{s} n + \frac{1}{\ln n + k} \sum_{j=1}^{n^*} \frac{1}{j} - 1.$$

Presupunem acum că atît funcția $p(j)$ cît și funcția $f(j)$ urmează legea lui Zipf. Expresia lui h va fi aceeași fiindcă numai $p(j)$ s-a schimbat și h nu este funcție de $p(j)$. În acest caz timpul de răspuns este

$$\frac{t_r}{t_s} = \sum_{j=1}^n \left[\frac{s}{c} \frac{1}{\ln n + k} \frac{1}{j} \right] \frac{1}{\ln n + k} \frac{1}{j}.$$

Aproximînd $\sum_{j=1}^n 1/j^2$ cu $\pi^2/6$ ca mai înainte se obțin limite ale raportului t_r/t_s valabile pentru toți n cu restricția $t_r/t_s \geq 1$

$$\frac{s}{c} (\ln n + k)^{-2} \frac{\pi^2}{6} \leq \frac{t_r}{t_s} \leq \frac{s}{c} (\ln n + k)^{-2} \frac{\pi^2}{6} + 1.$$

Ca și mai înainte, cînd $1 \leq n^* \leq n$ pot fi găsite noi limite. Deoarece pentru $j > n^*$,

$$\left[\frac{s}{c} \frac{1}{\ln n + k} \frac{1}{j} \right] = 1,$$

rezultă

$$\frac{t_r}{t_s} = \frac{1}{\ln n + k} \sum_{j=1}^{n^*} \left[\frac{s}{c} \frac{1}{\ln n + k} \frac{1}{j} \right] \frac{1}{j} + \sum_{j=1+n^*}^n \frac{1}{\ln n + k} \frac{1}{j}$$

și deci

$$\frac{s}{c} (\ln n + k)^{-2} \sum_{j=1}^{n^*} \left(\frac{1}{j^2} - \frac{c}{s} (\ln n + k) \right) + 1 \leq \frac{t_r}{t_s} \leq$$

$$\leq \frac{s}{c} (\ln n + k)^{-2} \sum_{j=1}^{n^*} \frac{1}{j^2} + 1.$$

În figura 11 este arătată comportarea raportului t_r/t_s , în cazul când $f(j)$ și $p(j)$ urmează legea lui Zipf. Mai jos vom considera alegerea

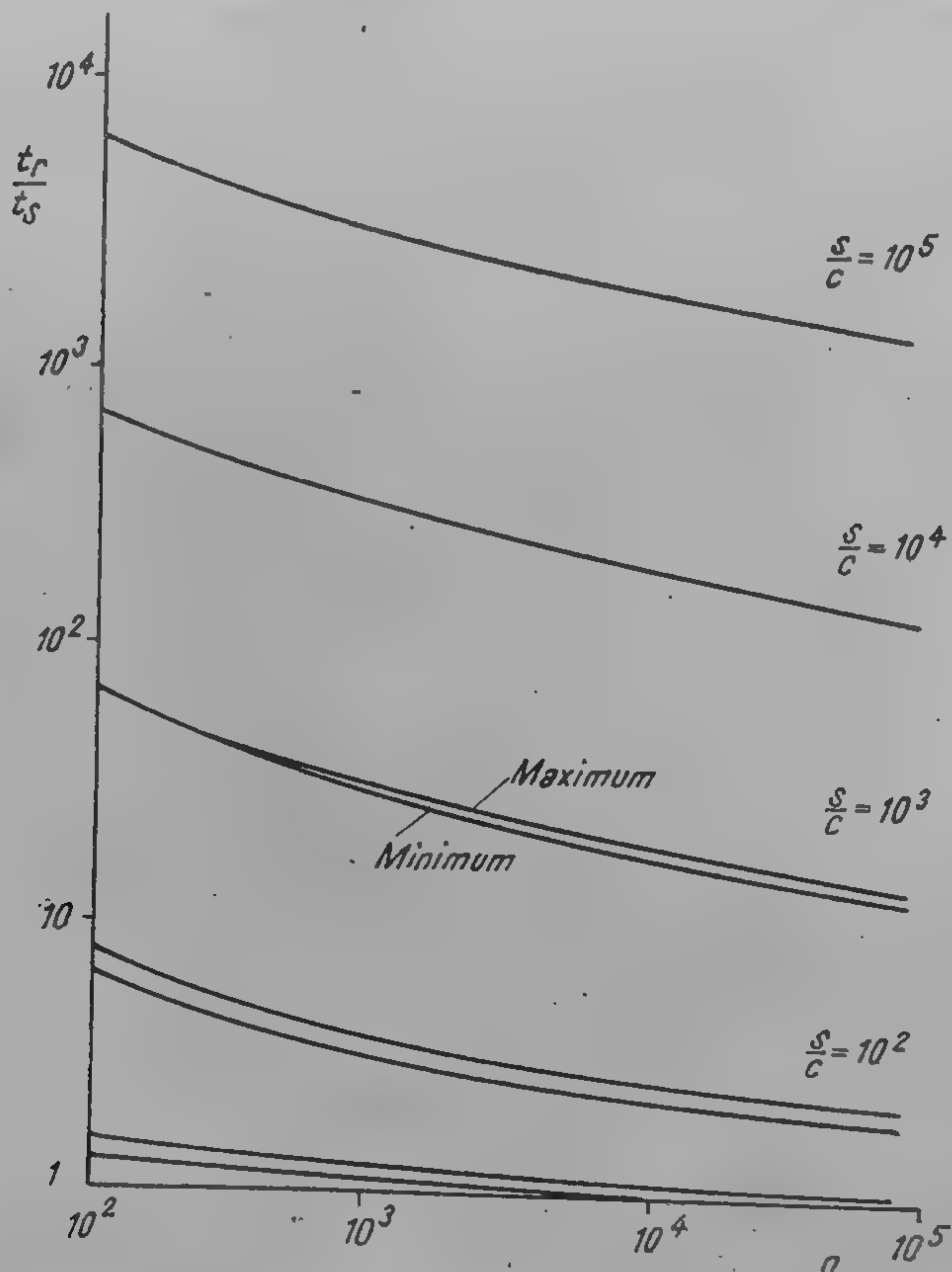


Fig. 11

unei valori a capacității unei locații c pentru o colecție cu 10^4 descriptori, în care acești descriptori apar de 10^6 ori. În acest caz $n = 10^4$, $s = 10^6$. Pentru diverse valori ale capacității c se pot obține diverse limite ale factorului de utilizare a memoriei și a timpului de răspuns normalizat. Cîteva valori tipice sînt ilustrate mai jos :

c	$\frac{h}{f}$ uniform	$\frac{h}{f}$ Zipf	$\frac{t_r/t_s}{f}$ uniform p uniform	$\frac{t_r/t_s}{f}$ Zipf p uniform	$\frac{t_r/t_s}{f}$ Zipf p Zipf
100	1,0—0,0	0,7—0,6	2,0—1,0	1,8—1,7	170—170
200	2,0—1,0	1,7—1,6	1,5—1,0	1,3—1,3	87—86
300	3,0—2,0	2,7—2,6	1,3—1,0	1,2—1,2	58—58
400	4,0—3,0	3,6—3,5	1,2—1,0	1,2—1,1	44—43
500	5,0—4,0	4,6—4,5	1,2—1,0	1,1—1,1	35—35

Cele două coloane h pentru comportarea uniformă și pentru comportarea Zipf a funcției $f(j)$ indică spațiul potențial pierdut dacă nu este realizată împachetarea. În celelalte trei coloane este arătat efectul lui c asupra timpului mediu de răspuns pentru trei combinații diferite ale comportării funcțiilor $f(j)$ și $p(j)$. Folosind ecuațiile deduse mai sus și o presupunere corespunzătoare privind $f(j)$ și $p(j)$ se poate selecta o valoare pentru c și se poate lua o decizie privind împachetarea a mai mult decît o listă inversată într-o locație.

4.2.3. Memorii asociative

Obiectul oricărui sistem de regăsire este să identifice locul unei informații dorite, făcînd comparația între informația stocată în memorie și cererea de selecție.

Deoarece sistemele de regăsire în urma comparației furnizează adrese, în ultimul timp se încearcă folosirea unor memorii care să lucreze după principiul identificării cuvintelor. Acest tip de memorie se numește cu adresare după conținut.

Dacă comparația se face simultan asupra tuturor înregistrărilor memorate, atunci memoria este cu căutare paralelă. Memoriile cu căutare paralelă trebuie să aibă deci asociate cu fiecare locație dispozitive pentru a permite informației stocate acolo să intre în

comparație în același timp cu informația din toate celelalte locații, însă independent de ea, și pentru a înregistra rezultatul comparației pentru acea locație.

Procesul de selecție constă din comparația fiecărui descriptor din registrul vectorului cererii de selecție cu fiecare descriptor al unei înregistrări stocate în memorie. Indicatori de coincidență indică coincidența dintre înregistrare și cererea de selecție și efectuează citirea datelor indentificate.

Dacă construcția memoriei este astfel că la fiecare selecție se folosesc toți descriptorii, avem de a face cu memorii catalog. Memoriile care permit să fie efectuată o selecție folosind numai anumiți descriptori se numesc asociative.

Pentru acest tip de memorie pentru cererea de selecție nu este adecvată reprezentarea printr-un vector logic — adică o secvență de 1 și 0 — deoarece este nevoie să se specifice pozițiile descriptorilor. Cererea de selecție trebuie reprezentată fie de un vector ale cărui componente sînt variabile ternare (1, 0 și „indiferent”), fie de o pereche de vectori logici. Ultima alternativă este cea folosită curent: pozițiile descriptorilor sînt date de un vector mască, iar valorile descriptorilor de un vector argument. Un zero în vectorul mască va indica că poziția respectivă nu trebuie folosită la selecție.

În figura 12 este dată schema bloc a unei memorii asociative.

Domeniul de stocare este reprezentat de o matrice M . Fiecare rînd M^* reprezintă o înregistrare. Matricea M are asociați trei vectori q , m și s . Primii doi vectori au aceleași dimensiuni cu rîndurile matricei și reprezintă registre care conțin argumentul și masca de selecție. Cel de-al treilea vector s , numit vector de identificare, reprezintă un registru care poate fi folosit fie pentru a comanda interogarea, fie pentru a comanda citirea rîndurilor.

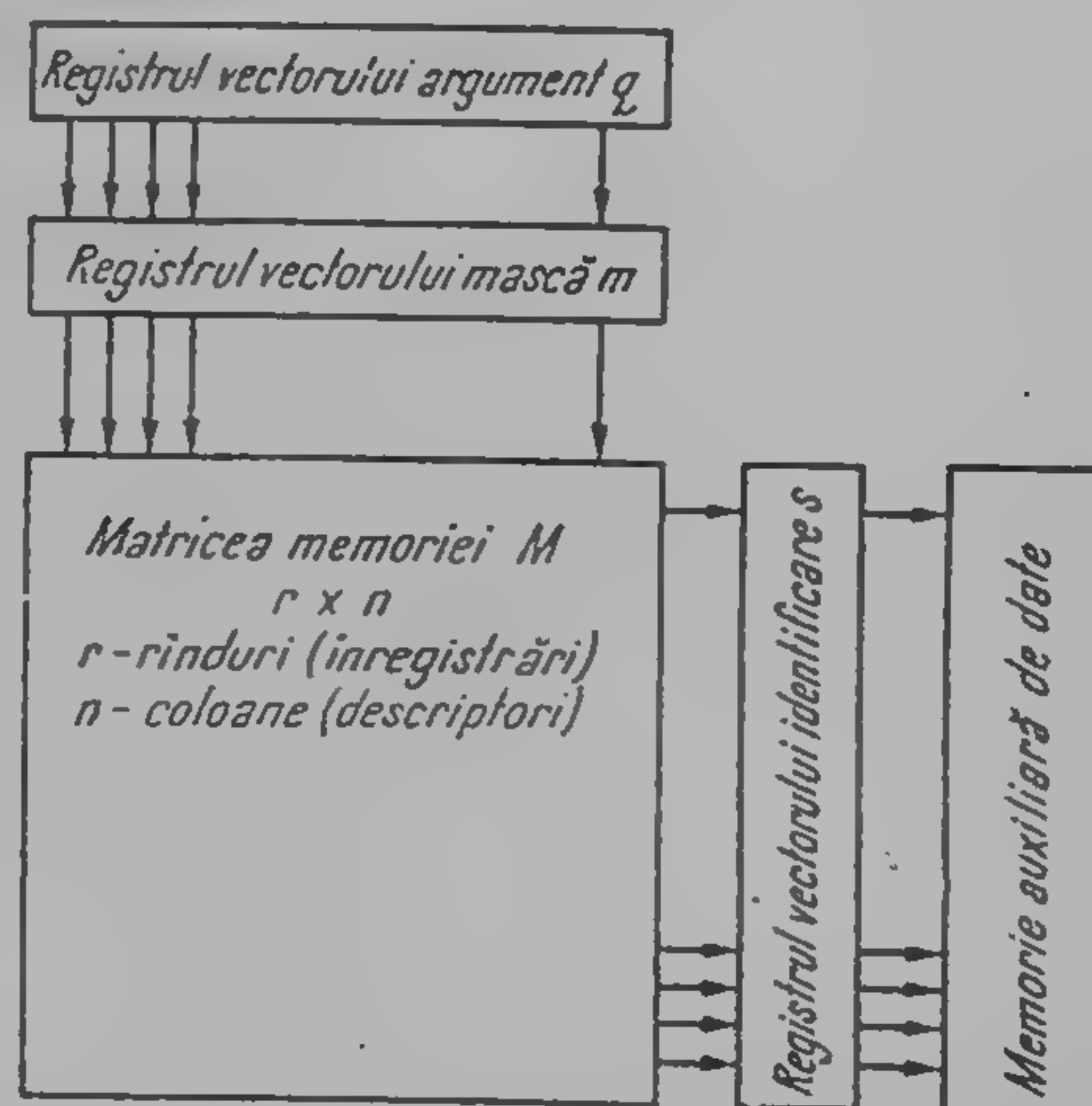


Fig. 12

O componentă a lui s corespunde fiecărui rînd al matricei și cifra 1 în s la sfîrșitul căutării va indica că locația corespunzătoare este identificată. Pentru a folosi scheme bloc a memoriei din figura 12, va fi necesar să specificăm metodele de comparare și mijloacele pentru stabilirea vectorului de identificare. De obicei memoriile de acest tip au un circuit de comparare în fiecare poziție de cifră binară. Aceasta înseamnă de fapt că valorile componentelor vectorului q sînt proiectate prin coloanele lui M și comparate cu informația corespunzătoare. Ca rezultat al acestei comparații se constituie o nouă matrice pe ale cărei rînduri să face o operație ulterioară pentru a obține componentele lui s .

Rîndul k al matricei de ieșire după comparare ar putea fi specificat*) ca $(M^k = q)$ sau complementul $(M^k \neq q)$ depinzînd de circuitul folosit.

În primul caz componenta corespunzătoare a vectorului s va fi specificată de operatorul „ȘI” aplicat tuturor componentelor rîndului

$$s_k \leftarrow \bigwedge / (M^k = q),$$

*) *Notăție*: Se folosește limbajul de programare elaborat de Iverson [68]. În acest limbaj toate operațiile aritmetice și logice definite în mod normal pentru scalari se extind sistematic la vectori și matrice. Astfel pentru vectorii a și b și orice operație \odot relația $c = a \odot b$ implică că c este vectorul definit de $c_i = a_i \odot b_i$. De exemplu $(1, 1, 0, 0) \wedge (1, 0, 1, 0) = (1, 0, 0, 0)$.

În al doilea rînd, operațiile logice obișnuite (ȘI, SAU, ...) sînt completate de afirmații relaționale. Dacă a și b sînt cantități oarecare și R este o relație definită peste ele atunci $(a R b)$ înseamnă variabila logică a cărei valoare este 1 dacă există relația și zero altfel. De exemplu $(4 > 3)$ are valoarea 1.

Expresia $(M^k \neq q_i)$ denotă un vector logic, c , astfel că $c_i = 1$ dacă $M_i^k \neq q_i$ și $c_i = 0$ dacă $M_i^k = q_i$. Dacă M^k și q sînt vectori logici, operația identitate este un „SAU exclusiv” între ei, componentă cu componentă.

Pentru operații care apar frecvent se definesc simboluri speciale. Astfel \odot/a este o „reducere” a vectorului a prin operația \odot . De exemplu, $+/ (4, 9, 1) = 14$, $\vee / (1, 0, 1, 1) = 1$ și $\wedge / (1, 0, 1, 1) = 0$. Linia diagonală apare de asemenea între doi vectori, ca u/a , unde u este un vector logic. Aceasta denotă o operație de „compresie” care elimină din a acele componente care corespund zerourilor din u . Astfel $(1, 0, 1, 0) / (4, 9, 1, 7) = (4, 1)$. Două alte simboluri de interes deosebit sînt $/a, u, b/$ și $\backslash a, u, b \backslash$, unde a și b sînt vectori și u este un vector logic. Aceste simboluri reprezintă „mascare” și „intercalare”

$$\begin{aligned} / (6, 7, 8), (1, 0, 1), (3, 4, 3) / &= (3, 7, 3) \\ \backslash (6, 7, 8), (1, 0, 0, 1, 0), (3, 4) \backslash &= (3, 6, 7, 4, 8). \end{aligned}$$

În sfîrșit, va fi notat că programele în acest limbaj sînt aranjate ca rețele verticale de afirmații cu săgeți auxiliare pentru a arăta ramificarea.

O afirmație $X \leftarrow Y$ se citește „ X este specificat de Y ”.

iar în al doilea caz de operatorul „SAU”

$$\bar{s}_k \leftarrow \vee / (M^k \neq q).$$

În ambele cazuri $s_k = 1$ indică coincidența vectorului linie M^k cu vectorul cerere q .

În cazul când se dorește numărarea componentelor 1 în fiecare înregistrare

$$s_k \leftarrow + / (M^k = q)$$

și componentele vectorului s pot fi ordonate după valorile s_k .

Sînt cîteva moduri în care poate fi folosită informația din registrul vectorului m însă în toate cazurile rolul ei este să schimbe funcția care comandă combinarea unei componente din matricea M cu o componentă a vectorului q . Astfel în logica de primul tip de mai sus, unde

$$s_k \leftarrow \wedge / (M^k = q),$$

masca ar putea fi introdusă după una din următoarele metode :

$$(a) s_k \leftarrow \wedge / (m / M^k = m / q),$$

$$(b) s_k \leftarrow \wedge / (m / (M^k = q)),$$

$$(c) s_k \leftarrow \wedge / (\backslash 1 \varepsilon, m, (m / M^k = m / q) \backslash),$$

$$(d) s_k \leftarrow \wedge / (1 \varepsilon, m, (M^k = q) /).$$

Rezultatul final al acestor expresii este același, însă fiecare metodă sugerează o realizare diferită a echipamentului. În (a) și (b) operatorul „ȘI” se aplică unui număr variabil de componente, ceea ce constituie o sarcină grea pentru proiectarea echipamentului. Metoda (a) specifică de asemenea că acolo unde $m_i = 0$ nu are loc nici o comparație, în timp ce metoda (b) specifică efectuarea de comparații asupra tuturor componentelor pentru ca apoi să se negligeze acele componente unde $m_i = 0$. În (c) și (d) intrările la operatorul „ȘI” pot rămîne fixe ca număr, însă conform metodei (c) acolo unde comparațiile au fost inhibitate se injectează 1 în timp ce, conform metodei (d), comparațiile nedorite sînt sărite.

Pentru a combina doi vectori conform unei operații logice, dispozitivul logic folosit trebuie să fie reprodus de un număr de ori egal cu numărul de componente al vectorilor. De aceea, operația logică este reprezentată ca un vector de operatori.

Fie ε un vector cu toate componentele 1. Reprezentînd operatorul de identitate cu 1 și operatorul de comparație cu $=$ și folosind produsul lor cu vectorul ε pentru a reprezenta vectori de operatori ale căror componente sînt în mod uniform 1 sau $=$, un vector de operatori p poate fi specificat ca

$$p \leftarrow \backslash 1\varepsilon, m, = \varepsilon \backslash.$$

Vectorul m a fost folosit aici pentru intercalare, adică o componentă a vectorului p este aleasă din 1ε , cînd $m_i = 0$, și din $= \varepsilon$, cînd $m_i = 1$. Funcționarea unei măști este descrisă deci de un program cu două linii, obținut prin combinarea liniei precedente cu linia

$$s_k \leftarrow \wedge / (M^k pq).$$

Acest program sugerează o realizare a echipamentului în care masca alege între funcții disponibile permanente în timp ce operatorul „ȘI” are un număr fix de intrări în toate condițiile.

Așa cum a fost definit un vector de operatori poate fi definită o matrice de operatori.

Mai jos sînt prezentate cîteva programe tipice pentru memorii asociative.

Programul 1 descrie funcționarea unei memorii cu căutare complet paralelă, în care o căutare de egalitate se face totdeauna într-o singură interogare. Acest program se bazează pe o logică cu operatori \neq și \vee cu toate că în funcționarea strict paralelă operatorii $=$ și \wedge lucrează la fel de bine.

1 →	$q \leftarrow q^*$
2	$m \leftarrow m^*$
3	$s \leftarrow 1\varepsilon$
4	$Q \leftarrow 1\varepsilon \times q$
5	$P \leftarrow \neq \varepsilon \times m$
6	$\bar{s} \leftarrow \vee / (MPQ) \vee \bar{s} \rightarrow$

Programul 1

Primele două linii ale programului specifică încărcarea registrelor q și m cu valorile date q^* și m^* . Linia 3 specifică stabilirea stării inițiale a registrului de identificare prin injectarea lui ε . În linia 4 proiecția valorilor componentelor lui q în matricea M este

reprezentată printr-o nouă matrice de referință Q ale cărei rînduri sînt toate identice cu q . În linia 5 cu o operație analogă se dezvoltă matricea operator P cu rînduri obținute după m , cu \neq acolo unde $m_i = 1$ și cu operatorul identic zero acolo unde $m_i = 0$. În final, linia 6 reprezintă determinarea vectorului s . Se observă că linia 3 ar fi putut fi omisă și linia 6 simplificată la

$$s \leftarrow \bigvee / (MPQ),$$

însă prin această echipamentul se complică, deoarece este mai greu să se injecteze 0 și 1 într-un registru alb decît să se repună la 1 registrul la începutul unui ciclu și apoi să se modifice numai într-o direcție.

Memoriile cu funcționare complet paralelă descrise de programul 1 sînt greu de realizat, deoarece elementele memoriei cu miezuri magnetice au raport semnal-zgomot foarte mic. Dezavantajul este înlăturat dacă memoria lucrează succesiv după descriptor. În principiu într-o astfel de memorie căutările de egalitate vor fi mai lente fiindcă fiecare căutare va cere un șir de interogări. Totuși în acest fel, după fiecare interogare starea vectorului s este disponibilă pentru scopuri logice.

Programul 2 descrie funcționarea unei memorii succesiv după descriptor folosind același tip de circuite ca și în cazul programului 1 și un echipament aproape identic. Cele două programe diferă însă prin utilizarea vectorului m .

În programul 1 vectorul m era folosit pentru a modifica aranjamentul static înainte de realizarea singurei interogări din linia 6. În programul 2 vectorul m este folosit pentru a comanda secvențial interogările după cum este arătat în linia 8.

1	→	$q \leftarrow q^*$	
2		$m \leftarrow m^*$	
3		$j \leftarrow 0$	
4		$s \leftarrow 1\varepsilon$	
5		$Q \leftarrow 1\varepsilon \times^* q$	=
6	→	$j : n$	→
7		$j \leftarrow j + 1$	
8	=	$m_j : 0$	
9	—	$\bar{s} \leftarrow \bar{s} \vee (M_j \neq Q_j)$	

Programul 2

Apare astfel o ramificație condițională care sare interogarea dacă componenta curentă a lui m este zero.

Programul 3 descrie de asemenea funcționarea unei memorii succesiv după descriptor însă după o metodă diferită. Există acum o matrice S formată din coloanele matricei M , coloana S_1 precedând coloana M_1 , iar coloana S_{n+1} succedând coloanei M_n .

Coloanele matricei S reprezintă vectori de indicație, vectorul S_{n+1} fiind echivalent cu vectorul s din programul precedent.

După cum se vede din linia 9 a acestui program, fiecare vector de indicație arată prin zerourile sale acele rînduri ale matricei M care au avut cel puțin un dezacord cu vectorul q , pînă la acel punct, și limitează interpretările ulterioare la acele rînduri care încă concordă.

1	→	$q \leftarrow q^*$	
2		$m \leftarrow m^*$	
3		$Q \leftarrow 1\epsilon \times q$	
4		$j \leftarrow 0$	
5		$S_1 \leftarrow 1\epsilon$	=
6	→	$j : n$	→
7		$j \leftarrow j + 1$	
8	—	$m_j : 0$	=
9		$S_{j+1} \leftarrow S_j \wedge S_j' (S_j / (M_j = Q_j))$	←
10	—	$S_{j+1} \leftarrow S_j$	

Programul 3

Coloana k a matricei M poate fi mascată mutînd înainte starea lui S_k în S_{k+1} , așa cum este arătat în ramificația de la linia 8 la linia 10. Procedeu este identic cu cel aplicat la linia 8 în programul 2, în sensul că starea lui S , după ce a fost interogată coloana k , este transmisă mai departe neschimbată dacă m_{k+1} este zero. În programul 2 informația este transmisă înainte, în timp ce în programul 3 este transmisă înainte și în spațiu.

Prima soluție este realizată mai ușor cu miezuri magnetice, semnalele fiind foarte apropiate ca mărime, iar a doua soluție este realizată mai ușor cu criotroni.

Cele trei programe prezentate mai sus sînt similare în ceea ce privește compararea directă a matricei M cu matricea Q . Vectorul q poate fi însă folosit pentru a modifica matricea M ; matricea mo-

dificată este comparată apoi cu vectorul q . Acest principiu este ilustrat în programul 4.

1	→	$q \leftarrow q^*$	
2		$a \leftarrow 0$	
3	→	$M \leftarrow \sqrt{q/M}, q, q/M \setminus$	→
4		$a : 1$	
5		$s \leftarrow \wedge / M$	
6	—	$a \leftarrow 1$	

Programul 4

Modificarea matricei M are loc în linia 3: oriunde q are un zero, coloana corespunzătoare a matricei M este complementată; în noua matrice acele rînduri care concordă cu q vor avea numai componente 1. Această stare de lucruri poate fi ușor sesizată în linia 5.

După determinarea vectorului s matricea M trebuie adusă la starea inițială. Această restabilire este realizată sub comanda alternatorului a .

4.3. OBSERVAȚII BIBLIOGRAFICE

Problema organizării colecției în memorii este prezentată după Warheit [186] și Meadow [99].

O comparație a colecțiilor organizate secvențial cu cele organizate în dicționar este făcută de Curtice [30].

Paragrafele privind organizarea în listă și organizarea în dicționar se bazează pe lucrările lui Lowe [88].

Paragraful privind memoriile asociative are la bază lucrările lui Falkoff [41] Kraizmer și colaboratori [75].

5 SISTEME CU CLASIFICARE AUTOMATĂ

Complexitatea procesului de regăsire a informațiilor depinde în mare parte de localizarea fizică a înregistrărilor în memorie. În general, într-un sistem bazat pe calculator timpul de prelucrare este o funcție monoton crescătoare de timpul de acces la înregistrarea specificată din memorie. Fiecare timp individual de acces este la rîndul său o funcție monoton nedescrescătoare de distanța relativă a fiecărei perechi de înregistrări la care se ajunge secvențial în structura memoriei.

Astfel o micșorare a timpului de interogare ar putea fi realizată simplu grupînd înregistrările care probabil sînt dorite împreună (de exemplu în același pachet de discuri sau pe aceeași rolă de bandă magnetică).

Problema grupării înregistrărilor este de primă importanță în sistemele de regăsire a informațiilor. Astfel în ultimii ani s-a observat un efort pentru rezolvarea problemei organizării unei mulțimi de înregistrări în scopul identificării unor submulțimi în așa fel încît într-o submulțime înregistrările să „semene” una cu alta și să „nu semene” cu înregistrări din afara submulțimi. Caracterul vag al termenului „seamănă” a împiedicat mult timp găsirea unui model matematic. Natura calitativă a relațiilor dintre elementele unei colecții mari este reflectată în adoptarea termenului „grupare” în loc de submulțime, implicînd astfel identificarea intuitivă a unui nucleu și o anumită libertate în definirea limitelor.

5.1. MATRICEA DE SIMILITUDINE ÎNTRE ÎNREGISTRĂRI

5.1.1. Matricea S

Considerăm matricea de fixare

$$F = \begin{bmatrix} v_{11} & v_{21} & \dots & v_{n1} \\ v_{12} & v_{22} & \dots & v_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ v_{1m} & v_{2m} & \dots & v_{nm} \end{bmatrix}.$$

Așa cum s-a definit similitudinea între coloanele acestei matrice se poate defini similitudinea între liniile matricei, adică între înregistrări.

Fie o funcție de distanță pe mulțimea $X \times X$, adică aplicația

$$\delta : X \times X \rightarrow R,$$

care fixează fiecărei perechi ordonate (x_j, x_k) un număr real astfel ca

$$[\forall x_j] [\forall x_k] (\delta(x_j, x_k) = \delta(x_k, x_j)),$$

$$[\forall x_j] [\forall x_k] (x_j = x_k \Leftrightarrow \delta(x_j, x_k) = 0),$$

$$[\forall x_i] [\forall x_j] [\forall x_k] (\delta(x_i, x_k) \leq \delta(x_i, x_j) + \delta(x_j, x_k));$$

în acest caz spațiul X devine un spațiu metric (X, δ) și funcția de distanță poate avea una din formele :

$$\delta_1(x_j, x_k) = \begin{cases} 0 & \text{dacă } x_j = x_k, \text{ adică } [\forall i] (d_i(x_j) = d_i(x_k)), \\ 1 & \text{dacă } x_j \neq x_k, \text{ adică } [\forall i] (d_i(x_j) \neq d_i(x_k)), \end{cases}$$

$$\delta_2(x_j, x_k) = \max |d_i(x_j) - d_i(x_k)|,$$

$$\delta_3(x_j, x_k) = \sqrt{\sum_{i=1}^n (d_i(x_j) - d_i(x_k))^2}.$$

Fie o funcție de apropiere pe mulțimea $X \times X$, adică aplicația

$$\alpha : X \times X \rightarrow R,$$

care fixează fiecărei perechi ordonate (x_j, x_k) un număr real astfel ca :

$$[\forall x_j] [\forall x_k] (\alpha(x_j, x_k) = \alpha(x_k, x_j)),$$

$$[\forall x_j] [\forall x_k] (x_j = x_k \Leftrightarrow \alpha(x_j, x_k) = 1).$$

Funcția α poate avea una din formele

$$\alpha_C(x_j, x_k) = \frac{\sum_{i=1}^n d_i(x_j) d_i(x_k)}{\left(\sum_{i=1}^n d_i^2(x_j) \cdot \sum_{i=1}^n d_i^2(x_k) \right)^{1/2}},$$

$$\alpha_{\text{PRN}}(x_j, x_k) = \frac{\sum_{i=1}^n d_i(x_j) d_i(x_k)}{\sum_{i=1}^n d_i^2(x_j) + \sum_{i=1}^n d_i^2(x_k) - \sum_{i=1}^n d_i(x_j) d_i(x_k)},$$

$$\alpha_S(x_j, x_k) = \frac{\sum_{i=1}^n \min(d_i(x_j), d_i(x_k))}{\min\left(\sum_{i=1}^n d_i(x_j), \sum_{i=1}^n d_i(x_k)\right)}.$$

Cu ajutorul funcțiilor α sau δ se poate construi o matrice de similitudine între înregistrări. Aceasta este o matrice pătrată simetrică $S = (s_{ij})$.

5.1.2. Matricea K

Considerăm mulțimea D a descriptorilor și J mulțimea primilor n întregi pozitivi.

Fie D_k mulțimea k -uplelor ordonate de elemente ale D . Fiecare $d' \in D_k$ este o submulțime de descriptori,

$$d' = \{d_1, \dots, d_k\},$$

unde $d_i \in D$ pentru fiecare $i = 1, \dots, k$.

Fie P o mulțime nevidă astfel încât fiecare $p \in P$ este o funcție definită pe D_k pentru $k \in J$ cu valori în $\{0, 1\}$. P este numită mulțimea predicatelor și se notează

$$P = \{p_{kj} \mid k \in J, j \in J\},$$

unde pentru fiecare $k, j \in J$, p_{kj} este cel de al j -lea predicat definit pe D_k .

Dacă $p_{kj} \in P$, atunci mulțimea

$$E(p_{kj}) = \{d' \mid d' \in D_k, p_{kj}(d') = 1\}$$

este numită extinderea predicatului p_{kj} .

O propoziție L este orice mulțime nevidă $\{d^i, p_{ki}\}$ astfel ca $d^i \in D_k, p_{ki}(d^i) = 1$ pentru orice $k \in J$.

Un document este o mulțime finită de propoziții.

Dacă $d^i, d^j \in D_k$, atunci d^i, d^j se zic conectați dacă și numai dacă există $p_{ki} \in P$ astfel ca

$$p_{ki}(d^i) = p_{ki}(d^j) = 1,$$

adică

$$d^i, d^j \in E(p_{ki});$$

d^i și d^j se zic conectați în documentul t dacă și numai dacă există k perechi de propoziții $L_{1r}, L_{2r}, r \in J$, încît

$$L_{1r} = \{d^i, p_r\}, L_{1r} \in t,$$

$$L_{2r} = \{d^j, p_r\}, L_{2r} \in t.$$

Scriem $d^i k d^j$ pentru a nota condiția definită mai sus.

Fie

$$k_i(d^i, d^j) = \max_k \{k | d^i k d^j, d^i \in L_j, d^j \in N_j, L_j, N_j \in t, j = 1, \dots, k\}.$$

Documentele t_1 și t_2 sînt k -conectate dacă și numai dacă

$$k \leq \sum_{i \in J_1} \sum_{j \in J_2} k_i^{12} U_i(d^i, d^j),$$

unde

$$J_1 = \{i | d^i \in L, L \in t\},$$

$$J_2 = \{j | d^j \in N, N \in t\}.$$

Scriem $t_1 k_{12} t_2$.

Valoarea k_{12} este un element al matricei de similitudine între înregistrări $K = (k_{ij})$.

5.2. METODA PREDISPOZIȚIILOR

Fiecărei perechi de înregistrări i se poate asocia un număr real pozitiv care reprezintă valoarea intensității de similitudine între înregistrări sau mai pe scurt valoarea similitudinii. Numim densitate a similitudinilor raportul dintre suma similitudinilor și numărul de perechi de înregistrări considerate.

Metoda predispozițiilor are la bază următoarele două postulate :

- Densitatea similitudinilor în interiorul oricărei submulțimi care constituie o grupare trebuie să fie nu mai mică decât media pe toate submulțimile care nu intersectează acea submulțime.
- Densitatea similitudinilor între submulțimea care constituie o grupare și complementul său trebuie să nu fie mai mare decât media pe toate submulțimile care intersectează acea submulțime.

În conformitate cu cele de mai sus se pot da următoarele definiții :

— O submulțime G a unei mulțimi X este o semigrupare dacă suma similitudinilor oricărui element al grupării G cu celelalte elemente din G depășește suma similitudinilor cu toate elementele din X care nu sînt în G .

— O semigrupare G la care nu poate fi adăugat nici un element din X fără să înceteze de a mai fi semigrupare este o grupare.

Definim predispoziția $b(x, G)$ a unui element x la o grupare G excesul (pozitiv sau negativ) similitudinii sale totale cu G față de similitudinea sa totală cu $X - G$. Putem spune atunci că o submulțime G este o grupare dacă toate elementele ei au predispoziție pozitivă cu G și toate elementele din afara lui G au predispoziție negativă cu G .

$$G = \{x \mid b(x, G) \geq 0\},$$

unde

$$b(x, G) = s(x, G) - s(x, X - G).$$

Termenul

$$s(x, G) = \sum_{g \in G} s(x, g)$$

reprezintă similitudinea totală a oricărei înregistrări x cu toate înregistrările $g \in G$.

Termenul

$$s(x, X - G) = \sum_{h \in X - G} s(x, h)$$

reprezintă similitudinea totală a oricărei înregistrări x la toate înregistrările $h \in X - G$.

Un procedeu posibil de grupare constă în alegerea unui element $g_i \in G$ și definirea unei mulțimi inițiale G_i formată cu elementul g_i și toate celelalte elemente care au o similitudine nenulă cu g_i . Este calculată apoi predispoziția $b(x, G_i)$ pentru toate elementele $x \in X$ și elementele cu predispoziție pozitivă sînt incluse în G_i , dacă nu sînt deja acolo. După fiecare transfer toate predispozițiile sînt recalculate și procesul este repetat pînă cînd nu mai are loc nici un transfer. G_i obținut astfel constituie o grupare și întregul procedeu este repetat pentru următorul element g_{i+1} .

Procedeul este convenabil pentru că este exhaustiv. Pentru același motiv însă calculele necesare pentru a separa o grupare sînt foarte lungi și eficiența procesului depinde în mare măsură de cît de aproape sînt mulțimile inițiale de gruparea finală.

O altă problemă o pune mărimea grupărilor. Dacă grupările devin prea mari se pot folosi metode de reducere pentru a elimina din fiecare grupare acele elemente a căror predispoziție este mai mică decît un prag. De fiecare dată cînd un element este eliminat toate predispozițiile trebuie recalculate și procedeul iterat pînă nu mai apar schimbări.

Problema principală rămîne însă găsirea unor mulțimi inițiale cît mai apropiate de gruparea finală.

Reprezentăm o submulțime cu f elemente printr-un vector w cu r componente, o componentă w_i avînd valoarea 1 sau -1 corespunzător faptului că o înregistrare x_i este sau nu membră a grupării. Un astfel de vector îl vom numi vector vîrf, deoarece reprezintă vîrfurile unui hiper cub de mărime 2.

Fie S_i un vector linie din matricea S . Atunci predispoziția unui element $x_i \in X$ la o grupare G este numărul

$$S_i w = \sum_{j=1}^r s_{ij} w_j.$$

Rezultă că un vector vîrf asociat cu o grupare are proprietatea că

$$S w = Q w,$$

unde S este matricea de similitudine, iar Q este o matrice diagonală nenegativă. Elementele matricei Q sînt modulii predispozițiilor și ecuația spune că componentele pozitive ale vectorului w corespund predispozițiilor pozitive și invers.

Un vector, vîrf sau nu, care satisface ecuația va fi numit stabil la semn pentru matricea S .

Căutarea de grupări corespunde astfel căutării de vectori vîrf stabili la semn pentru matricea de similitudine.

Vectorii proprii ai matricei S care corespund valorilor proprii pozitive sînt astfel de vectori stabili la semn.

Matricea S pătrată și simetrică este reductibilă dacă rîndurile și coloanele ei pot fi permutate astfel încît să poată fi constituite submatrice formate numai din zerouri, adică dacă poate fi pus în forma cvasidiagonală

$$S^* = \begin{bmatrix} \boxed{S_1} & 0 \dots 0 & 0 \\ 0 \dots 0 & \boxed{S_2} & 0 \dots 0 \\ 0 \dots 0 & 0 \dots 0 & \boxed{S_n} \end{bmatrix},$$

unde S_1, \dots, S_n sînt submatrice pătrate.

Notînd cu P matricea permutărilor, atunci matricele S și S^* sînt asemenea dacă

$$S = P^t S^* P.$$

Fie $\lambda_1, \lambda_2, \dots, \lambda_n$ valorile proprii și u_1, u_2, \dots, u_n vectorii proprii ai matricei S^* . Considerăm matricea ortogonală U dată de

$$U = (u_1, u_2, \dots, u_n),$$

care este formată folosind vectorii proprii u_i drept coloane. Transpusa matricei U este

$$U^t = \begin{bmatrix} u_1^t \\ u_2^t \\ \vdots \\ u_n^t \end{bmatrix},$$

iar forma canonică Jordan a matricei S^* este

$$S^* = U \Delta U^t$$

și

$$S = P^t S^* P = P^t U \Delta U^t P.$$

Notînd

$$\Delta' = P' \Delta P$$

se poate scrie

$$S = (P'UP)\Delta' (P'UP)'$$

Vectorii proprii și valorile proprii ale matricei S sînt aceleași cu cele ale matricei S^* . Componentele vectorilor proprii indică pozițiile elementelor submatricelor încît vectorul propriu poate fi folosit ca un vector de selecție al elementelor unei grupări.

O submatrice construită astfel reprezintă o grupare inițială ideală în care fiecare element este legat de celelalte elemente ale grupării și nu are nici o legătură cu elementele din afara grupării. În cazul unei matrice reale de similitudine se poate întîmpla să nu existe zerouri.

Vom analiza acum cazul cînd în matricea simetrică toate elementele nule se înlocuiesc cu valori mici. Proprietățile acestei matrice se scot din proprietățile matricei cu zerouri, presupunînd că se cunosc valorile proprii și vectorii proprii ai acesteia.

Considerăm matricea reală de similitudine

$$S + \varepsilon S_0$$

unde S_0 este o matrice simetrică.

Valorile proprii ale matricei $S + \varepsilon S_0$ sînt funcții de ε . Se poate arăta că pentru $\varepsilon \rightarrow 0$ valorile proprii și vectorii proprii ai matricei $S + \varepsilon S_0$ tind spre valorile și vectorii proprii ai matricei S .

Fie $\lambda_1(\varepsilon), \lambda_2(\varepsilon), \dots, \lambda_n(\varepsilon)$ valorile proprii și $u_1(\varepsilon), \dots, u_n(\varepsilon)$ vectorii proprii ai matricei $S + \varepsilon S_0$. Se poate arăta că $\lambda_k(\varepsilon)$ și $u_k(\varepsilon)$ sînt funcții continue și derivabile de ε unde $\lambda_k(0) = \lambda_k$ și $u_k(0) = u_k$.

Reprezentăm aceste funcții sub forma

$$\lambda_k(\varepsilon) = \lambda_k + \varepsilon \lambda_k^1 + \dots,$$

$$u_k(\varepsilon) = u_k + \varepsilon u_k^1 + \dots$$

și vom căuta întîi pe λ_k^1 și u_k^1 , adică partea principală a corecției. Obținem

$$(S + \varepsilon S_0) u_k(\varepsilon) = \lambda_k(\varepsilon) u_k(\varepsilon),$$

adică

$$(S + \varepsilon S_0)(u_k + \varepsilon u_k^1 + \dots) = (\lambda_k + \varepsilon \lambda_k^1 + \dots)(u_k + \varepsilon u_k^1 + \dots).$$

Egalind termenii de grad 1 în raport cu ε în ambii membrii ai egalității se obține

$$Su_k^1 + S_0 u_k = \lambda_k u_k^1 + \lambda_k^1 u_k.$$

Înmulțim scalar ambii membrii ai egalității cu u_k

$$(Su_k^1, u_k) + (S_0 u_k, u_k) = \lambda_k (u_k^1, u_k) + \lambda_k^1 (u_k, u_k).$$

Deoarece matricea S este simetrică și deci

$$(Su_k^1, u_k) = (u_k^1, Su_k) = \lambda_k (u_k^1, u_k),$$

avem

$$(S_0 u_k, u_k) = \lambda_k^1 (u_k, u_k) = \lambda_k^1.$$

Pentru calculul corecției u_k^1 înmulțim scalar ambii membrii ai egalității cu u_i , unde $i \neq k$. Deoarece vectorii u_k și u_i sînt ortogonali, adică $(u_k, u_i) = 0$, pentru $i \neq k$ obținem

$$(Su_k^1, u_i) + (S_0 u_k, u_i) = \lambda_k (u_k^1, u_i).$$

Însă analog cu cazul precedent avem

$$(Su_k^1, u_i) = (u_k^1, Su_i) = \lambda_i (u_k^1, u_i)$$

și de aceea

$$(u_k^1, u_i) = \frac{(S_0 u_k, u_i)}{\lambda_k - \lambda_i}.$$

Alegem ca bază vectorii proprii u_1, \dots, u_n ai matricei S ,

$$(Su_j, u_i) = c_{ij}.$$

Coordonatele vectorului u_k^1 le notăm a_1, \dots, a_n , adică

$$u_k^1 = a_1 u_1 + \dots + a_n u_n,$$

ceea ce înseamnă că

$$a_i = (u_k^1, u_i).$$

Coordonatele de rang k se determină din condiția de normare a vectorului propriu u_k , adică din condiția ca lungimea vectorului $u_k + \varepsilon u_k^1 + \dots$ să fie 1,

$$((u_k + \varepsilon u_k^1 + \dots), (u_k + \varepsilon u_k^1 + \dots)) = 1,$$

adică

$$(u_k, u_k) = \varepsilon((u_k^1, u_k) + (u_k, u_k^1)) + \dots = 1.$$

Egalînd termenii de gradul 1 în ε , obținem

$$(u_k^1, u_k) + (u_k, u_k^1) = 0.$$

Acastă condiție poate fi satisfăcută punînd

$$a_k = (u_k^1, u_k) = 0.$$

În sfîrșit obținem

$$u_k^1 = \sum_{i \neq k} \frac{c_{ik}}{\lambda_k - \lambda_i} u_i,$$

unde $c_{ik} = (S_0 u_k, u_i)$, iar λ_k sînt valorile proprii ale matricei S .

5.3. METODA GRAFELOR NEORIENTATE

Mulțimea X a înregistrărilor,

$$X = \{x_1, x_2, \dots, x_r\},$$

poate fi descrisă de un graf neorientat \mathcal{A} . În acest graf o latură l_{hk} între vîrfurile x_h și x_k există dacă și numai dacă în matricea de similitudine S există $s_{hk} > 0$.

Graful \mathcal{A} este complet descris de matricea sa adiacentă $r \times r$ $A = (a_{hk})$, unde $a_{kh} = a_{hk} = 1$ dacă și numai dacă în matricea de similitudine $s_{hk} > 0$.

Considerăm o rețea unidimensională Y cu r celule care poate fi pusă în corespondență cu o mulțime de indici $1, 2, \dots, r$.

O fixare τ a lui \mathcal{A} este o transformare a grafului \mathcal{A} în rețeaua Y astfel că dacă (j_1, j_2, \dots, j_r) este o permutare a întregilor $(1, 2, \dots, r)$, înregistrarea x_{j_i} este fixată la celula i și notăm

$$x_{j_i} \rightarrow i.$$

Fiind dată o fixare generică τ , presupunem $a_{hk} = 1$ și $x_h \rightarrow i_h$, $x_k \rightarrow i_k$. Numim scurtare relativă a laturii l_{hk} prin fixarea τ mărimea

$$z_{hk} = |i_h - i_k|.$$

Deci pentru fiecare fixare mărimea

$$Z = \frac{1}{2} \sum_{h,k=1}^r a_{hk} z_{hk},$$

numită scurtare relativă totală, este definită și calculabilă.

Vom introduce unele funcții definite pe mulțimea celulelor rețelei și vom nota ψ^j valoarea funcției ψ la celula j .

Numim grad al x_{j_i} în \mathcal{A} numărul g_{j_i} de înregistrări conectate direct la x_{j_i} din care la b^i le-au fost fixate celule cu indici mai mari decît i iar la c^i le-au fost fixate celule cu indici mai mici decît i :

$$g_{j_i} = b^i + c^i.$$

Cu alte cuvinte b^i este numărul de laturi care pleacă din i spre dreapta, iar c^i este numărul de laturi care pleacă din i spre stînga, considerînd că celulele $1, 2, \dots, r$ sînt aranjate în ordine naturală.

Pentru fiecare celulă introducem funcția incrementală z^i

$$z^i = b^i - c^i$$

și funcția cumulativă f^i

$$f^i = \sum_{j=1}^i z^j,$$

care dă numărul de legături interceptate la o secțiune între i și $i+1$, adică numărul de legături care merg la dreapta din celulele $1, 2, \dots, i$, mai puțin numărul de legături care se termină pe celulele $2, 3, \dots, i$.

Deoarece $z_{j_h, j_k} = |h - k|$ dă o contribuție de o unitate la $f^h, f^{h+1}, \dots, f^{k-1}$ în cazul că $h < k$ și deoarece $f^r = 0$, adică nici o legătură nu există la dreapta celulei r , rezultă

$$Z = \sum_{j=1}^{r-1} f^j.$$

Fie un nod generic x_{j_i} conectat la $x_{j_{h_1}}, x_{j_{h_2}}, \dots, x_{j_{h_s}}$. Prin fixarea τ ,

$$x_{j_{h_s}} \rightarrow h_s.$$

Definim funcția potențială a nodului x_{j_i} prin

$$\rho_i^j = \sum_{r=1}^s |j - h_r|.$$

Cu alte cuvinte ρ_i^j reprezintă suma scurtărilor relative ale tuturor legăturilor conectate la x_{j_i} dacă x_{j_i} este plasat în celula j fără să afecteze fixarea vreunui alt nod.

Pentru o fixare dată τ pentru orice celulă j astfel ca

$$h_r \leq j < h_{r+1}$$

creșterea lui ρ_i^j este dată de

$$\rho_i^{j+1} - \rho_i^j = r - (s - r),$$

adică diferența dintre numărul de legături conectate la $x_{j_{h_{r+1}}}$, ..., $x_{j_{h_s}}$ și numărul de legături conectate la $x_{j_{h_1}}$, ..., $x_{j_{h_r}}$. De fapt deplasarea lui x_{j_i} cu o poziție la dreapta face ca scurtările legăturilor conectate la prima mulțime să crească cu o unitate, în timp ce scurtările legăturilor conectate la cea de-a doua mulțime scad cu o unitate. Deci în intervalul $h_r \leq j < h_{r+1}$, ρ_i^j este o funcție liniară a cărei creștere este $2r - s$.

La fiecare celulă h_1, h_2, \dots, h_s creșterea funcției ρ_i^j pentru creșterea lui j înseamnă o discontinuitate pozitivă de două unități, fiindcă la fiecare celulă o legătură trece de la mulțimea din dreapta la mulțimea din stînga. Prin urmare, funcția ρ_i^j descrește dacă $2r - s < 0$ și crește dacă $2r - s > 0$.

Ca o concluzie a celor afirmate mai sus, putem spune că dacă x_{j_i} este conectat la $x_{j_{h_1}}, x_{j_{h_2}}, \dots, x_{j_{h_s}}$, atunci funcția ρ_i^j este o funcție liniară al cărei minim este atins la

$$h_{s/2} \leq j \leq h_{\frac{s}{2}+1} \quad \text{dacă } s \text{ e par,}$$

$$h_{\frac{s+1}{2}} = j \quad \text{dacă } s \text{ e impar.}$$

În fine, decurgînd direct din definiție, se observă că

$$Z = \frac{1}{2} \sum_{i=1}^r \rho_i^j.$$

Vom considera acum problema schimbării fixării τ într-o altă fixare τ' . Operația de bază utilizată în acest scop este permutarea ciclică dreaptă. Dacă nodurile $x_{j_r}, x_{j_{r+1}}, \dots, x_{j_s}$ sînt fixate respectiv la celulele $r, r+1, \dots, s$ după realizarea permutării ciclice drepte, pe care o vom nota (s/r) , ele vor fi fixate respectiv la celulele $r+1, r+2, \dots, s, r$. Evident, orice fixare poate fi obținută din oricare altă fixare printr-un număr finit de permutări ciclice drepte. De fapt, orice fixare este o permutare, fiecare permutare este egală cu un număr finit de transpoziții, fiecare transpoziție este echivalentă cu un număr finit de permutări ciclice drepte.

Vom căuta acum o expresie pentru variația lui Z determinată de o permutare ciclică dreaptă. În primul rînd trebuie observat că o permutare ciclică dreaptă (s/r) rezultă din realizarea succesivă a dislocării ($x_{j_r} \rightarrow r+1, x_{j_{r+1}} \rightarrow r+2, \dots, x_{j_{s-1}} \rightarrow s$) și din inserarea $x_{j_s} \rightarrow r$. Vom examina acum efectul fiecărei operații asupra lui Z .

Considerăm dislocarea și următoarele mulțimi de legături:
 A_{rs} = mulțimea legăturilor de la $(1, 2, \dots, r-1)$ la $(s, s+1, \dots, m)$,

B_{rs} = mulțimea legăturilor de la $(1, 2, \dots, r-1)$ la $(r, r+1, \dots, s-1)$,

C_{rs} = mulțimea legăturilor de la $(r, r+1, \dots, s-1)$ la $(s, s+1, \dots, m)$.

Fie a_{rs}, b_{rs}, c_{rs} numerele cardinale ale mulțimilor A_{rs}, B_{rs} și C_{rs} . Dislocarea nu afectează scurtările legăturilor lui A_{rs} în timp ce scurtările tuturor legăturilor lui B_{rs} cresc cu o unitate și scurtările lui C_{rs} descresc cu o unitate.

Deci variația ΔZ_1 a lui Z datorită numai dislocării este

$$\Delta Z_1 = b_{rs} - c_{rs}.$$

Deoarece

$$f^{r-1} = b_{rs} + a_{rs},$$

$$f^{s-1} = c_{rs} + a_{rs},$$

atunci

$$\Delta Z_1 = f^{r-1} - f^{s-1}.$$

Considerăm acum inserția $x_{j_s} \rightarrow r$. Variația ΔZ_2 a lui Z datorită acestei operații va fi $\rho_r' - \rho_r$ dacă dislocarea nu a avut loc. Este necesară deci o corecție. Scurtarea fiecărei legături din s la

$(r, r+1, \dots, s-1)$ apare ca redusă cu 1 în ΔZ_1 , în timp ce în realitate trebuie să crească cu 1 ca efect al permutării ciclice drepte. Deci dacă sînt $v_{r,}$ astfel de legături schimbarea totală a lui Z este

$$\Delta Z = \Delta Z_1 + \Delta Z_2 + 2 v_{r,} = (f^{r-1} + \rho_r') - (f^{s-1} + \rho_r') + 2 v_{r,}.$$

După realizarea unei permutări ciclice drepte (s/r) valorile lui f^j sînt modificate (pentru $r < j < s$).

Fie $v_{r,} = k$ și fie s legat cu i_1, i_2, \dots, i_k astfel ca $r < i_1 < i_2 < f < \dots < i_k < s$. Notăm cu f'^j valorile lui f^j după realizarea permutării ciclice drepte (s/r) . Există atunci următoarele relații:

$$f'^j = f^j \text{ pentru } 1 \leq j < r, s < j \leq m,$$

$$f'^j = f^{j-1} + (f^s - f^{s-1}) + 2k \text{ pentru } r \leq j < i_1,$$

$$f'^j = f^{j-1} + (f^s - f^{s-1}) + 2(k-m) \text{ pentru } i_m < j \leq i_{m+1}, m=1, 2, \dots, k-1,$$

$$f'^j = f^{j-1} + (f^s - f^{s-1}) \text{ pentru } i_k < j < s.$$

Pe baza tuturor celor de mai sus se poate elabora un algoritm pentru reducerea funcției Z .

De fapt dacă funcția f^j este cunoscută, calculînd pe ΔZ se poate stabili imediat dacă o permutare ciclică dreaptă propusă va conduce la o micșorare a lui Z : $\Delta Z < 0$ va constitui regula de decizie pentru executarea acestei permutări. În al doilea rînd, funcția f^j poate fi modificată simplu cu ajutorul expresiilor care dau pe f'^j .

Fiind dată o fixare τ a grafului \mathcal{A} , adică o transformare $x_{j_i} \rightarrow i$, unde x_{j_i} este o înregistrare a colecției și i o celulă a rețelei Y , se pot construi următoarele tabele cu m intrări:

T_1 în care intrarea i conține celula lui Y în care este memorat x_i ;

T_2 în care intrarea j conține înregistrarea memorată în celula j a rețelei Y . T_2 este inversul lui T_1 ;

T_3 în care intrarea j conține valoarea curentă a funcției f^j ;

T_4 în care intrarea h conține lista tuturor înregistrărilor legate în graf la x_h .

Cu ajutorul acestor patru tabele se poate construi următorul algoritm pentru reducerea scurtării relative totale Z :

1. Se pune $j = 2$.
2. Se caută intrarea j a lui T_2 ; fie aceasta x_h .
3. Se caută intrarea h a lui T_4 și se obțin toate înregistrările legate la x_h .

4. Din T_1 se obțin celulele în care sînt înmagazinate înregistrările obținute la 3.

5. Cu ajutorul lui T_3 se calculează

$$\psi^m = f^{m-1} + \rho_j^m + 2 v_{jm}$$

pentru $m = j, j-1, j-2, \dots$

Se găsește maximumul lui ψ^m . Fie acesta ψ^r .

6. Se formează $\Delta = \psi^j - \psi^r$. Dacă $\Delta \leq 0$ se merge la 7. Altfel se face (j/r) .

7. Se reface T_1, T_2 și T_3 .

8. Dacă $j = m$ stop. Altfel se înlocuiește j cu $j+1$ și se trece la 2.

Aplicarea algoritmului satisface cerința ca valoarea curentă a lui Z să fie monoton necrescătoare. Practic cu acest algoritm rețeaua Y este parcursă de la stînga la dreapta cu cîte o celulă în fiecare treaptă și se determină dacă înregistrarea conținută în ultima celulă parcursă poate fi adusă la stînga printr-o permutare cîclică dreaptă astfel încît Z să scadă. După ce Y este parcursă de la stînga la dreapta, urmează o parcurgere de la dreapta la stînga pentru a se rearanja înregistrările pentru care în timpul primei parcurgeri $\Delta Z > 0$. Se completează astfel un ciclu de prelucrare. Mai jos algoritmul este ilustrat cu un exemplu.

Fie graful din figura 13.

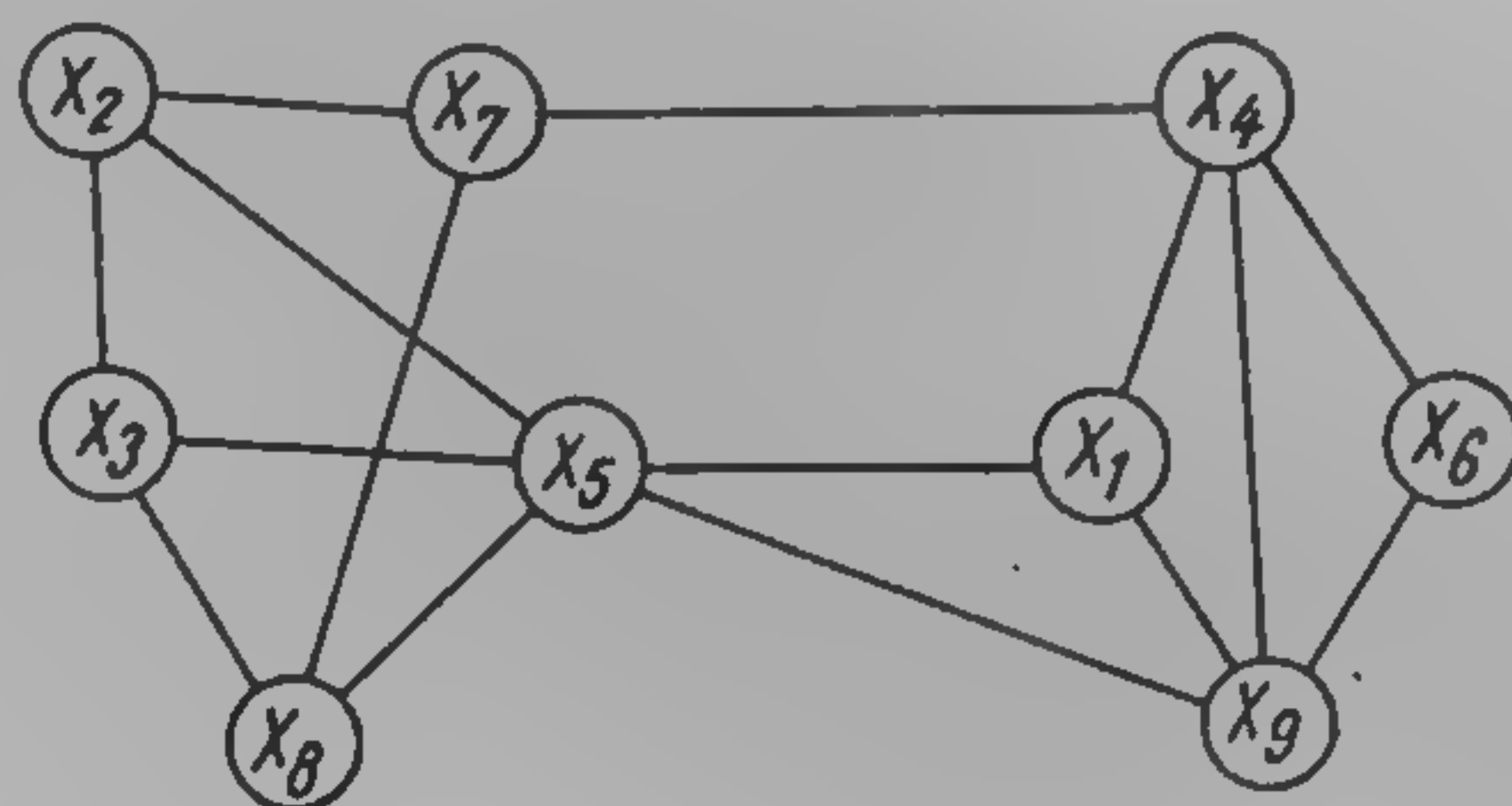


Fig. 13

În tabela 1 sînt date funcțiile z^j și f^j pentru fixarea inițială dată în figura 14. În acest caz $Z = 56$.

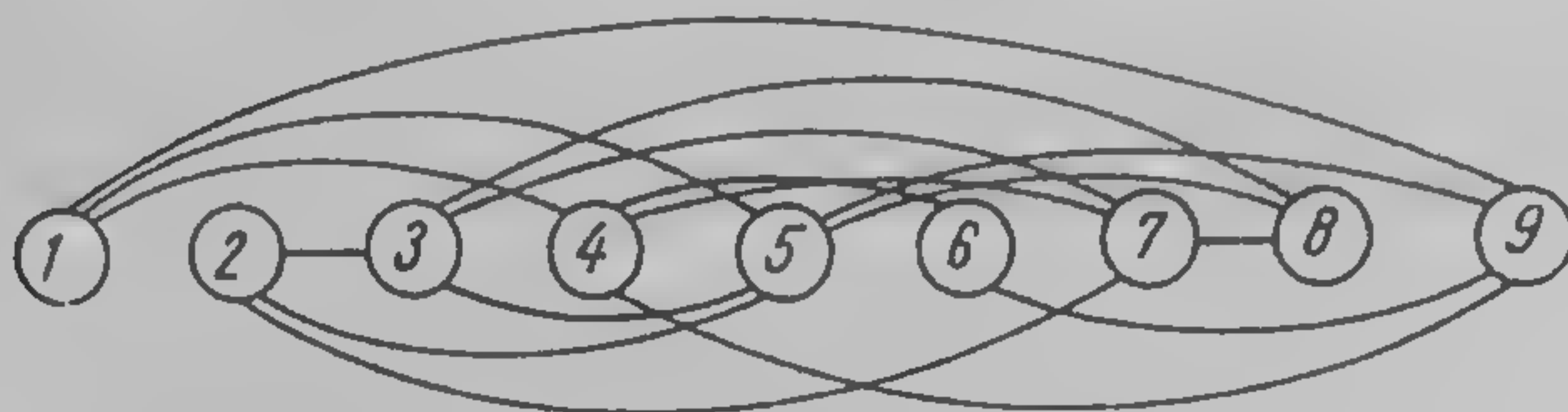


Fig. 14

Tabela 1

j	1	2	3	4	5	6	7	8	9
z'	3	3	2	2	-1	0	-2	-3	-4
f'	3	6	8	10	9	9	7	4	0

Aplicînd algoritmul, după o parcurgere de la stînga la dreapta se obține fixarea din figura 15.

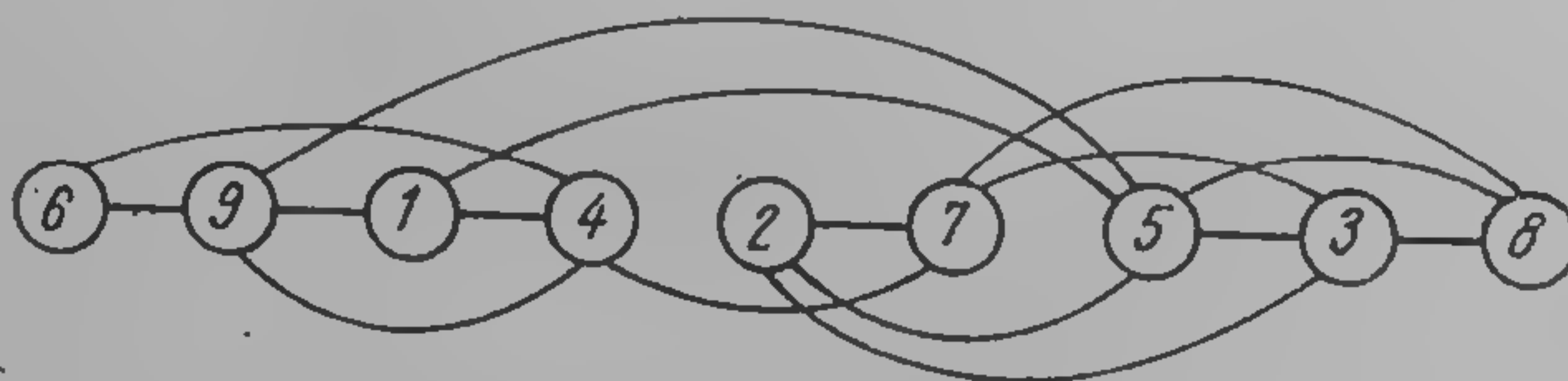


Fig. 15

În acest caz funcțiile z' și f' sînt cele din tabela 2.

Tabela 2

j	6	9	1	4	2	7	5	3	8
z'	2	2	1	-2	3	0	-1	-2	-3
f'	2	4	5	3	6	6	5	3	0

pentru care $Z = 34$.

După o a doua parcurgere (de la dreapta la stînga) se obține fixarea din figura 16 pentru care $Z = 30$.

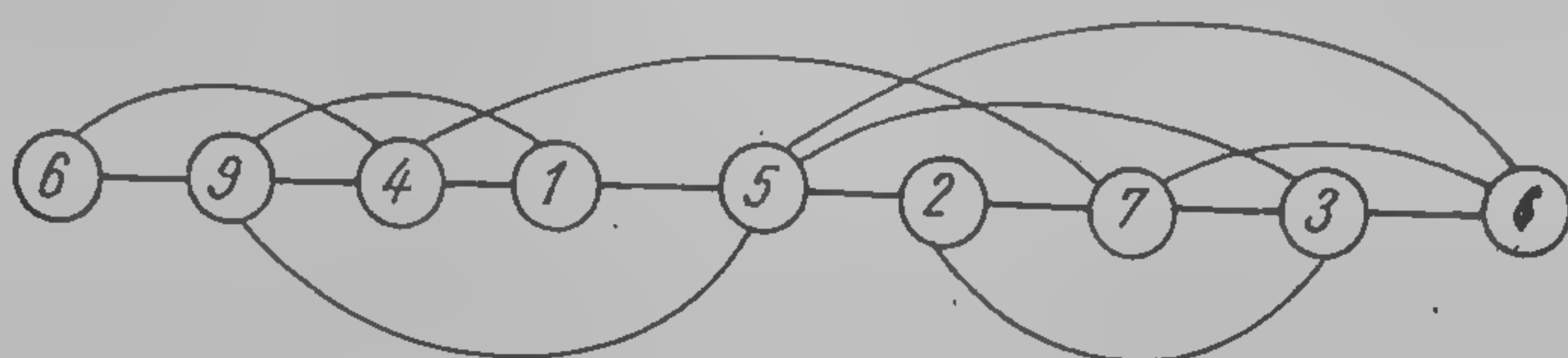


Fig. 16

Au fost identificate astfel două grupări într-un singur ciclu al algoritmului.

5.4. METODA DISTANȚEI MINIME

5.4.1 Funcții discriminante liniare

Fiecare înregistrare x , desemnată de vectorul

$$x = \{d_k(x) | k = 1, \dots, n\},$$

reprezintă un punct în spațiul n -dimensional numit spațiul înregistrărilor.

Clasificarea înregistrărilor poate fi definită și ca separarea unor submulțimi de puncte în acest spațiu cu ajutorul unor suprafețe de decizie. Dacă se face o clasificare în m grupări, atunci suprafețele de decizie împart spațiul înregistrărilor în m regiuni numite regiuni de decizie.

Suprafețele de decizie pot fi definite implicit cu m funcții scalare uniforme de înregistrarea $x: f_1(x), f_2(x), \dots, f_m(x)$. Aceste funcții pe care le vom numi funcții discriminante sînt alese astfel încît pentru toate înregistrările x în regiunea i

$$f_i(x) > f_j(x), \quad i, j = 1, \dots, m, i \neq j.$$

Cu alte cuvinte în regiunea i cea de-a i funcție discriminantă are cea mai mare valoare.

Presupunem de asemenea că funcțiile discriminante sînt continue de-a lungul suprafețelor de decizie. În acest caz suprafața

de decizie care separă regiunile contigue i și j este dată de

$$f_i(\omega) - f_j(\omega) = 0.$$

Clasificarea unei înregistrări se reduce în acest caz la calcularea a m funcții discriminante și selectarea acelei grupări care corespunde funcției cu valoare maximă.

O familie importantă de funcții discriminante este cea a funcțiilor liniare de forma

$$f(\omega) = \sum_{i=1}^n w_i d_i(\omega) + w_{n+1},$$

unde $d_i(\omega)$ sînt componentele vectorului ω , iar w_i sînt parametrii funcției.

În majoritatea cazurilor funcțiile discriminante se obțin printr-un proces de instruire cu înregistrări prototip ale căror componente reprezintă parametrii funcției discriminante folosite.

Presupunem că sînt date m mulțimi finite de puncte prototip X_1, X_2, \dots, X_m , fiecare mulțime X_i avînd t_i puncte $x_i^1, x_i^2, \dots, x_i^{t_i}$.

Definim distanța euclidiană $\delta(\omega, X_i)$ din punctul arbitrar ω la mulțimea X_i ca

$$\delta(\omega, X_i) = \min_j |\omega - x_i^j|, \quad j = 1, \dots, t_i,$$

adică distanța dintre ω și X_i este cea mai mică distanță dintre ω și fiecare punct în X_i .

Definim clasificarea de distanță minimă față de mulțimile X_1, X_2, \dots, X_m plasarea fiecărei înregistrări într-o grupare asociată cu cea mai mică distanță.

O clasificare echivalentă se obține comparînd patratele distanțelor. Deoarece

$$|\omega - x_i^j|^2 = (\omega - x_i^j)(\omega - x_i^j),$$

$$(\omega - x_i^j)^2 = \omega\omega - 2\omega x_i^j + x_i^j x_i^j,$$

clasificarea de distanță minimă poate fi efectuată comparînd expresiile

$$\omega x_i^j - \frac{1}{2} x_i^j x_i^j.$$

În acest caz pentru fiecare $i = 1, \dots, m$ definim funcțiile discriminante date de expresia

$$f_i(x) = \max f'_i(x), \quad j = 1, \dots, t_i,$$

unde f'_i este o funcție discriminantă subsidiară de forma

$$f'_i(x) = \sum_{k=1}^n d_k(x'_i) d_k(x) + \left(-\frac{1}{2} x'_i x'_i \right).$$

Pentru orice vector x , cea mai mare valoare o va lua funcția discriminantă al cărui index este asociat cu mulțime X_i , cea mai apropiată de x .

O astfel de clasificare este convenabilă dacă fiecare grupare este reprezentată de un număr redus de înregistrări prototip.

5.4.2. Clasificarea cu matrice instruibile

O matrice instruibilă reprezintă o rețea cu cuplaje condiționate între elemente ale căror funcții logice depind de semnalele aplicate anterior. Cu alte cuvinte o matrice instruibilă constituie un dispozitiv de comutare de structură matriceală la care legăturile funcționale sînt realizate cu elemente de legătură. Aceste elemente pot fi binare, în mai multe trepte și analoge. Elementul are canalele de lucru a și b care în funcție de operația îndeplinită pot servi atît ca intrări, cît și ca ieșiri ale lui. Proprietățile elementului sînt determinate de următoarele operații principale (fig. 17):

— *operația instruirii*: ambele canale ale elementului sînt intrări la care se aduc simultan semnale a și b . Între intrările a și b se stabilește o legătură $c = \theta_0 ab$, unde θ_0 este coeficientul de legătură;

— *operația ab* : la intrarea a se aduce semnalul de apel a , iar la ieșirea b se produce semnalul $s = \beta_0 ac$, unde β_0 este coeficientul de proporționalitate;

— *operația ba* : la intrarea b se aduce semnalul de apel b , iar la ieșirea a se produce semnalul $s = \beta_0 bc$.

Reunirea elementelor de legătură într-o structură matriceală constituie matricea instruibilă. În figura 18 se indică o astfel de structură reprezentată simbolic prin bare. Convenim să denumim

coloanele și liniile matricei prin a și b . Matricea instruibilă realizează următoarele operații:

— *operația de instruire*: pe barele verticale se aplică înregistrarea x sub forma a n semnale corespunzătoare descriptorilor d_1

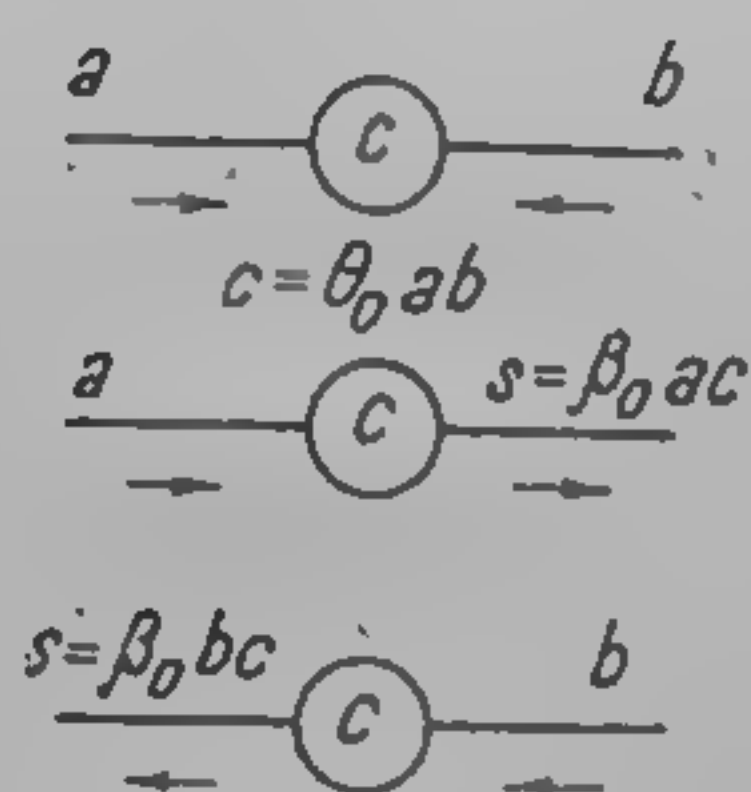


Fig. 17

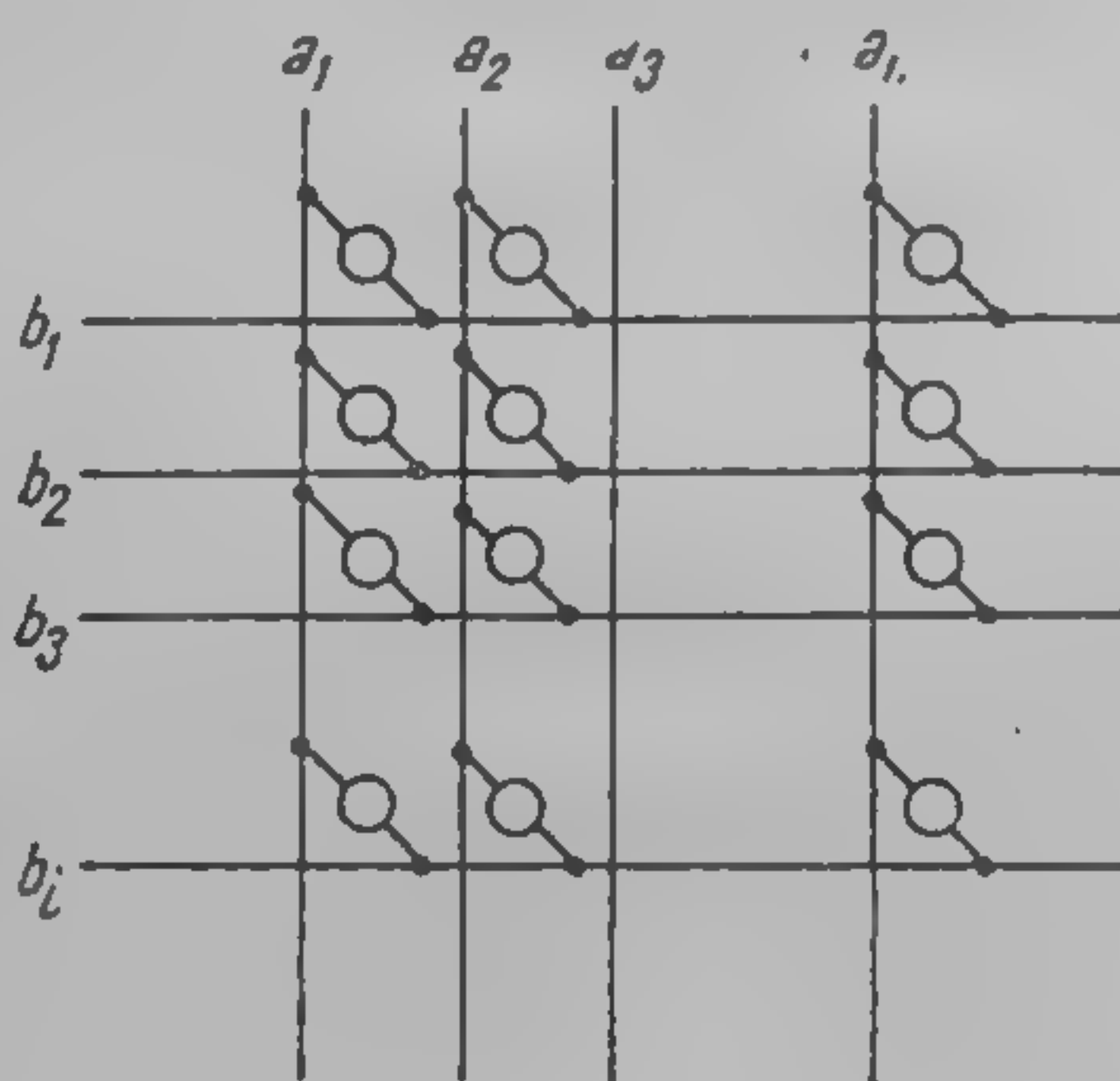


Fig. 18

d_2, \dots, d_n . În același timp la barele orizontale se aplică semnalele corespunzătoare liniei de înregistrare, astfel ca la un moment dat numai una din bare este excitată. Fiecare element de legătură al barei b_i îndeplinește operația de instruire și produce legătura

$$c_{ij} = \theta_0 d_j b_i, \quad j = 1, \dots, n.$$

Dacă considerăm legăturile pentru toate elementele liniei b_i , atunci putem defini legătura convențională c_i :

$$c_i = c_{i1}, \dots, c_{in} = \theta d_{11}, \dots, \theta d_{in},$$

unde $\theta = \theta_0 b_i = \text{const.}$ pentru toate elementele matricei. Legătura convențională c_i reprezintă înregistrarea codificată și prin urmare formarea ei este echivalentă cu memorarea înregistrării x ;

— *operația de clasificare*: pe barele a se aplică înregistrarea prototip sub forma a n semnale corespunzătoare descriptorilor săi $d_{p1}, d_{p2}, \dots, d_{pn}$. Fiecare element al matricei produce semnalul

$$\beta_0 c_{ij} d_{pj} = \beta_0 \theta d_i d_{pj}, \quad j = 1, \dots, n.$$

În mod corespunzător la ieșirea liniilor b se alcătuiesc semnalele totale

$$s_i = \beta_0 \theta \sum_{j=1}^n d_j d_{pj}.$$

Mai jos se va arăta că în anumite condiții găsirea lui $\max s_i$ este echivalentă cu găsirea lui $\min \delta(x, x_p)$, adică semnalul s_i de valoare maximă se produce în acea linie a matricei în care s-a memorat înregistrarea cea mai asemănătoare cu înregistrarea prototip.

Vom determina acum condiția ca

$$\min \delta(x, x_p) \rightarrow \max s_i.$$

Fie o înregistrare x reprezentată de vectorul

$$x = \{d_k(x) | k = 1, \dots, n\}$$

a cărui lungime este

$$|x| = \sqrt{\sum_{j=1}^n d_j^2}.$$

Distanța dintre acest vector și un vector prototip este

$$\delta(x, x_p) = \sqrt{\sum_{j=1}^n (d_j - d_{pj})^2}.$$

Convenim ca vectorii x să fie numiți normați dacă au aceeași lungime, adică

$$|x| = k.$$

Vectorii nenormați pot fi normați pentru un număr real anterior prescris k dacă fiecare înregistrare x s-ar transforma într-o înregistrare

$$x \frac{k}{|x|}.$$

Din expresia distanței se deduce

$$\min \delta(x, x_p) \rightarrow \min \sum_{j=1}^n d_j^2 - 2 \sum_{j=1}^n d_j d_{pj} + \sum_{j=1}^n d_{pj}^2.$$

Întrucît termenul $\sum_{j=1}^n d_{pj}^2 = |x_p|^2$ are o valoare constantă, este valabilă următoarea expresie :

$$\min \delta(x, x_p) \rightarrow \max \sum_{j=1}^n d_{ij} d_{pj} - \frac{1}{2} \sum_{j=1}^n d_j^2.$$

Pentru vectori normați expresia capătă forma

$$\min \delta(x, x_p) \rightarrow \max \sum_{j=1}^n d_i d_{pj} \rightarrow \max_i s_i.$$

Dacă vectorii nu sînt normați, atunci este necesar ca valoarea semnalului s_i să fie proporțională cu $\sum_{j=1}^n d_i d_{pj} - \frac{1}{2} \sum_{j=1}^n d_j^2$. Aceasta se poate realiza introducînd în matrice o coloană suplimentară de elemente de legătură d_{n+1} , în care pe durata operației de instruire se memorează mărimile $-\frac{1}{2} \sum_{j=1}^n d_j^2$. Dacă în timpul operației de clasificare la coloana d_{n+1} se trimite apelul cu un semnal unitar, atunci semnalul total produs în linia b_i va fi egal cu

$$s_i = \beta_0 \theta \left(\sum_{j=1}^n d_i d_{pj} - \frac{1}{2} \sum_{j=1}^n d_j^2 \right)$$

și deci

$$\min \delta(x, x_p) \rightarrow \max s_i.$$

Astfel, într-o matrice instruibilă îndeplinirea condiției de mai sus pretinde satisfacerea uneia din următoarele două condiții :

- matricea să posede o schemă de normare cu ajutorul căreia vectorii x se normează înaintea operației de instruire ;
- matricea să posede o coloană suplimentară d_{n+1} și o schemă de comandă, care să asigure pe durata operației de instruire producerea mărimilor $-\frac{1}{2} \sum_{j=1}^n d_j^2$ și memorarea lor în elementele coloanei d_{n+1} .

5.4.3. Caracterul invariant al clasificării

Un aspect important al problemei clasificării cu ajutorul matricei instruibile este acela al invarianței clasificării față de unele transformări din spațiul înregistrărilor.

Dacă T este una din acele transformări, atunci înregistrarea $T(x_p)$, obținută din x_p ca rezultat al aplicării T pe durata operației de clasificare, clasifică la fel ca și înregistrarea x_p . Cu alte cuvinte, invarianța înseamnă că există echivalența

$$\min \delta(x, x_p) \rightarrow \min \delta(x, T(x_p)).$$

O asemenea echivalență este posibilă atunci când

$$\max \sum_{j=1}^n d_j T(d_{pj}) - \frac{1}{2} \sum_{j=1}^n d_j^2 \rightarrow \max \sum_{j=1}^n d_j d_{pj} - \frac{1}{2} \sum_{j=1}^n d_j^2.$$

Prin urmare pentru a demonstra invarianța în fiecare caz concret, este suficient să se demonstreze că este îndeplinită această condiție.

Ca exemplu vom examina invarianța clasificării față de transformarea de similitudine $T(x_p) = kx_p$ și față de transformarea de deplasare $T(x_p) = x_p + kv$, unde $k > 0$ este o mărime scalară luată arbitrar, iar v este un vector arbitrar.

Clasificarea este invariantă față de transformarea similitudinii, dacă vectorii x sînt normați. Într-adevăr, pentru transformarea similitudinii condiția este îndeplinită numai atunci cînd termenul

$$- \frac{1}{2} \sum_{j=1}^n d_j^2 \text{ are o valoare constantă. Aceasta este posibil numai atunci}$$

cînd vectorii x sînt normați. Deci dacă se cere ca clasificarea să fie invariantă față de transformarea similitudinii, matricea trebuie prevăzută cu o schemă de normare.

Clasificarea este invariantă față de transformarea de deplasare atunci cînd tuturor înregistrărilor x le corespund în spațiul înregistrărilor puncte ce se găsesc pe același hiperplan

$$\sum_{j=1}^n d_j d_{vj} = r.$$

Pentru transformarea de deplasare

$$T(d_{pj}) = d_{pj} + kd_{vj}.$$

condiția de invarianță se realizează dacă $k: \sum_{j=1}^n d_j d_{v_j}$ are o valoare constantă, ceea ce înseamnă că tuturor înregistrărilor w le corespund puncte pe un același hiperplan. Desigur că în cazul general înregistrările nu satisfac această condiție și atunci ele trebuie transformate cu ajutorul expresiei

$$w + \frac{r - \sum_{j=1}^n d_j d_{v_j}}{\sum_{j=1}^n d_{v_j}^2} v,$$

unde r și v sînt date anterior.

5.4.4. Proprietatea corecției automate

Presupunem că distanța minimă între două înregistrări memorate x_i, x_j este $\delta_0 > 1$, iar înregistrarea prototip x_p este situată față de o înregistrare oarecare memorată x , la o distanță mai mică decît $\frac{\delta_0 - 1}{2}$ dacă δ_0 este un număr impar sau $\frac{\delta_0 - 2}{2}$, dacă δ_0 este un număr par. Vom arăta că în acest caz x este clasat univoc ca x_p . Pentru aceasta este suficient să se demonstreze că este satisfăcută inegalitatea

$$\delta(x_p, x_j) < \delta(x_i, x_p), \quad i \neq j.$$

Conform definiției distanței

$$\delta(x_i, x_p) + \delta(x_p, x_j) \geq \delta(x_i, x_j),$$

de unde

$$\delta(x_i, x_p) \geq \delta(x_i, x_j) - \delta(x_p, x_j),$$

$$\delta(x_i, x_p) \geq \min \delta(x_i, x_j) - \max \delta(x_p, x_j),$$

$$\delta(x_i, x_p) \geq \delta_0 - \frac{\delta_0 - 1}{2}.$$

Prin urmare

$$\delta(x_i, x_p) \geq \frac{\delta_0 + 1}{2},$$

$$\delta(x_p, x_i) \leq \frac{\delta_0 - 1}{2},$$

deci

$$\delta(x_p, x_i) < \delta(x_i, x_p).$$

Rezultatul obținut poate fi interpretat ca o proprietate de corecție automată, adică o invarianță a clasificării relativ la modificările nesistematice în descriptorii înregistrărilor.

5.5. METODA FUNCȚIILOR DE APARTENENȚĂ

5.5.1. Funcția de apartenență

Fie o mulțime X . Orice submulțime $G \subset X$ poate fi definită cu ajutorul funcției caracteristice

$$\chi_G : X \rightarrow R,$$

definită pentru fiecare element $x \in X$ astfel

$$\chi_G(x) = \begin{cases} 1 & \text{dacă } x \in G, \\ 0 & \text{dacă } x \notin G, \end{cases}$$

În mod similar putem introduce noțiunea de funcție de apartenență

$$\varphi_G : X \rightarrow R,$$

care asociază fiecărui $x \in X$ un număr real în intervalul $[0,1]$. Valorile $\varphi_G(x)$ reprezintă gradul de apartenență al înregistrării x la gruparea G .

Vom spune că o grupare este o submulțime $G \subset X$ caracterizată de o funcție de apartenență φ_G cu următoarele proprietăți:

$$[\forall x \in X] (\varphi_G(x) = 0) \Leftrightarrow G = \emptyset,$$

$$[\forall x \in X] (\varphi_1(x) < \varphi_2(x)) \Leftrightarrow G_1 \subset G_2,$$

$$G_3 = G_1 \cap G_2 \Leftrightarrow \varphi_3(x) = \min(\varphi_1(x), \varphi_2(x)),$$

$$G_4 = G_1 \cup G_2 \Leftrightarrow \varphi_4(x) = \max(\varphi_1(x), \varphi_2(x)).$$

5.5.2. Utilizarea funcțiilor de probabilitate ca funcții de apartenență

Fie $p(x/i)$ probabilitatea ca un element $x \in X$ să aparțină grupării i
 p_k^i probabilitatea ca un descriptor d_k să aparțină grupării i .

O funcție de apartenență se poate construi folosind probabilitățile $p(x/i)$ și p_k^i împreună cu situația conținutului de descriptori pentru fiecare înregistrare.

Considerăm m grupări și matricea

$$\begin{bmatrix} p_1^1 & p_2^1 & \dots & p_n^1 \\ p_1^2 & p_2^2 & \dots & p_n^2 \\ \dots & \dots & \dots & \dots \\ p_1^m & p_2^m & \dots & p_n^m \end{bmatrix},$$

unde fiecare linie corespunde unei grupări și fiecare coloană corespunde unui descriptor.

Fiecărei grupări îi corespunde un vector linie

$$W_i = \{p_k^i | k = 1, \dots, n\}.$$

Fie o înregistrare

$$x = \{d_k(x) | k = 1, \dots, n\}.$$

Vom spune că o familie de aplicații $\varphi_G : X \rightarrow R$ este parametrizată de mulțimea $\{W_1, W_2, \dots, W_m\}$ dacă aplicația

$$\varphi : X \times \{W_1, W_2, \dots, W_m\} \rightarrow R,$$

definită de

$$\varphi(x, W_i) = \varphi_{W_i}(x),$$

este continuă.

Pentru simplificare vom nota $\varphi_{W_i}(x) = \varphi_i(x)$.

Exemple de funcții de apartenență sînt următoarele funcții:

$$\varphi_{i1}(x) = \frac{\sum_{k=1}^n \min(p_k^i, d_k(x))}{\sum_{k=1}^n \max(p_k^i, d_k(x))},$$

$$\varphi_{i2}(x) = \frac{\sum_{k=1}^n \min(p_k^i, d_k(x))}{\min \sum_{k=1}^n p_k^i, \sum_{k=1}^n d_k(x)},$$

$$\varphi_{i3}(x) = \frac{\sum_{k=1}^n p_k^i d_k(x)}{\left(\sum_{k=1}^n (p_k^i)^2 \cdot \sum_{k=1}^n d_k^2(x) \right)^{1/2}},$$

$$\varphi_{i4}(x) = \frac{\sum_{k=1}^n p_k^i d_k(x)}{\sum_{k=1}^n (p_k^i)^2 + \sum_{k=1}^n d_k^2(x) - \sum_{k=1}^n p_k^i d_k(x)},$$

În cazul cînd vectorii înregistrare sînt binari, condițiile de funcție de apartenență sînt îndeplinite și de funcția

$$\varphi_{i5}(x) = \frac{N_i \prod_{k=1}^n p_k^i d_k(x)}{\sum_{j=1}^m N_j \prod_{k=1}^n p_k^j d_k(x)},$$

unde N_i este numărul de înregistrări în gruparea i , iar $d_k(x)$ sînt numai descriptorii nenuli ai unei înregistrări x .

Dacă introducem notațiile

$$\pi_k^i = \begin{cases} p_k^i & \text{dacă } d_k \in x, \\ 1 - p_k^i & \text{dacă } d_k \notin x, \end{cases}$$

funcțiile de mai sus pot fi scrise în forma lui Baker

$$\varphi_{i_s}(x) = \frac{N_i \prod_{k=1}^n \pi_k^i}{\sum_{j=1}^m N_j \prod_{k=1}^n \pi_k^j}.$$

Toate funcțiile de mai sus au ca parametrii probabilitățile descriptorilor de a aparține la o anumită grupare.

Aceste probabilități pot fi determinate fie folosind eşantioane de înregistrări prototip cu clasificare cunoscută apriori, fie folosind întreaga colecție și una din metodele analizei claselor latente. În ambele cazuri valorile p_k^i se pot calcula folosind frecvența relativă drept măsură cantitativă de estimare a posibilității obiective.

Astfel în primul caz avînd N înregistrări prototip împărțite în m categorii, fiecare categorie avînd N_i înregistrări în total și N_k^i înregistrări care au descriptorul d_k , atunci

$$p_k^i = \frac{N_k^i}{N_i}.$$

Această metodă nu pare a fi indicată pentru sistemele de regăsire a informațiilor, unde este greu sau imposibil să se determine înregistrări prototip. Pentru determinarea parametrilor p_k^i se poate folosi însă întreaga colecție.

Fie $p(x/i)$ probabilitatea ca o înregistrare $x \in X$ să aparțină grupării i . Vom nota cu p_z probabilitatea ca o înregistrare $x \in X$ să conțină descriptorii $z \subset D$ și cu p_z^i probabilitatea ca în gruparea i o înregistrare $x \in X$ să aibă descriptorii z .

Ținînd seama de independența statistică a descriptorilor

$$p_z^i = \prod_{k \in z} p_k^i.$$

În felul acesta se poate stabili ecuația fundamentală din analiza claselor latente

$$p_z = \sum_{i=1}^m p(x/i) p_z^i.$$

Valorile $p(x/i)$ și p'_i pot fi determinate cunoscând probabilitățile p_z și rezolvînd ecuația de mai sus. Valorile p_z pot fi calculate pentru o mulțime X dată considerîndu-le ca frecvențe relative. Astfel dacă N_z este numărul înregistrărilor cu descriptorii z și N este numărul total de înregistrări, atunci

$$p_z = \frac{N_z}{N}.$$

Metoda de rezolvare a ecuației fundamentale este dată în anexa 2.

5.5.3. Convexitatea grupărilor determinate de funcții de apartenență

Un segment de extremități x_1 și x_2 este o mulțime de elemente de forma

$$x_1 + (1 - \lambda)x_2,$$

unde λ este un număr real, astfel ca $0 \leq \lambda \leq 1$. O submulțime din spațiul X este convexă dacă include pentru fiecare pereche de elemente x_1 , x_2 segmentul de extremități x_1 și x_2 . Partea vidă a lui X este convexă prin definiție.

Spunem că o grupare G este convexă dacă și numai dacă mulțimile

$$A_c = \{x \mid \varphi_G(x) \geq c\}$$

sînt convexe pentru toți c în intervalul $[0, 1]$.

Dacă punem $c = \varphi_G(x_1) \leq \varphi_G(x_2)$, atunci datorită convexității lui A_c , $x_2 \in A_c$ și $\lambda x_1 + (1 - \lambda)x_2 \in A_c$. Deci se poate spune că o grupare este convexă dacă și numai dacă

$$\varphi_G(\lambda x_1 + (1 - \lambda)x_2) \geq c = \varphi_G(x_1) = \min(\varphi_G(x_1), \varphi_G(x_2)).$$

Dacă grupările G_i și G_j sînt convexe, atunci intersecția lor $G_k = G_i \cap G_j$ este convexă, adică

$$\varphi_k(\lambda x_1 + (1 - \lambda)x_2) = \min(\varphi_i(\lambda x_1 + (1 - \lambda)x_2), \varphi_j(\lambda x_1 + (1 - \lambda)x_2)).$$

Fiindcă G_i și G_j sînt grupările convexe

$$\varphi_i(\lambda x_1 + (1 - \lambda)x_2) \geq \min(\varphi_i(x_1), \varphi_i(x_2)),$$

$$\varphi_j(\lambda x_1 + (1 - \lambda)x_2) \geq \min(\varphi_j(x_1), \varphi_j(x_2)),$$

și deci

$$\varphi_k(\lambda x_1 + (1 - \lambda) x_2) \geq \min(\min(\varphi_i(x_1), \varphi_i(x_2)), \min(\varphi_j(x_1), \varphi_j(x_2))),$$

sau echivalent

$$\varphi_k(\lambda x_1 + (1 - \lambda) x_2) \geq \min(\min(\varphi_i(x_1), \varphi_j(x_1)), \min(\varphi_i(x_2), \varphi_j(x_2)))$$

și deci

$$\varphi_k(\lambda x_1 + (1 - \lambda) x_2) \geq \min(\varphi_k(x_1), \varphi_k(x_2)).$$

În cele ce urmează vom arăta că funcțiile descrise în §5.5.2 caracterizează grupări convexe.

Pentru a arăta că funcția $\frac{\sum_{k=1}^n p_k' d_k(x)}{n}$ caracterizează o grupare convexă va trebui demonstrat că

$$\frac{\sum_{k=1}^n (\lambda d_k(x_1) + (1 - \lambda) d_k(x_2)) p_k'}{n} \geq \min\left(\frac{\sum_{k=1}^n d_k(x_1) p_k'}{n}, \frac{\sum_{k=1}^n d_k(x_2) p_k'}{n}\right).$$

Notăm

$$\frac{\sum_{k=1}^n d_k(x_1) p_k'}{n} = a,$$

$$\frac{\sum_{k=1}^n d_k(x_2) p_k'}{n} = b,$$

$$\frac{\sum_{k=1}^n (\lambda d_k(x_1) + (1 - \lambda) d_k(x_2)) p_k'}{n} = c$$

și presupunem $a > b$.

Atunci

$$c = \lambda a + (1 - \lambda) b.$$

Dacă $\lambda = 0$,

$$o = b, \text{ deci } o = \min(a, b).$$

Dacă $\lambda = 1$,

$$o = a, \text{ deci } o > \min(a, b).$$

Dacă $\lambda = \frac{1}{\tau}$

$$o = \frac{a}{\tau} + \frac{\tau-1}{\tau} b$$

și deoarece $a > b$,

$$o > \frac{b}{\tau} + \frac{\tau-1}{\tau} b > b, \text{ deci } o > \min(a, b).$$

În mod similar se arată că celelalte funcții satisfac relația

$$\frac{\varphi_i(\lambda x_1 + (1-\lambda)x_2)}{\varphi_j(\lambda x_1 + (1-\lambda)x_2)} \geq \min\left(\frac{\varphi_i(x_1)}{\varphi_j(x_1)}, \frac{\varphi_i(x_2)}{\varphi_j(x_2)}\right).$$

cu condițiile

$$\varphi_i(\lambda x_1 + (1-\lambda)x_2) \geq \min(\varphi_i(x_1), \varphi_i(x_2)),$$

$$\varphi_j(\lambda x_1 + (1-\lambda)x_2) \geq \min(\varphi_j(x_1), \varphi_j(x_2)).$$

5.5.4. Separarea grupărilor

În spațiul X n -dimensional al înregistrărilor, un hiperplan

$$H = \{x \mid ax = o\}$$

determină trei mulțimi

$$X_1 = \{x \mid ax < o\},$$

$$X_2 = \{x \mid ax = o\},$$

$$X_3 = \{x \mid ax > o\},$$

pe care prin convenție le notăm H_- , H , H_+ .

Atît hiperplanul H , cît și semispațiile deschise H_- și H_+ sînt mulțimi convexe. În mod similar se definește separarea grupărilor convexe.

Fie G_1 și G_2 două grupări mărginite, h un număr dependent de H astfel ca $\varphi_1(x) \leq h$ pe o parte a lui H și $\varphi_2(x) \leq h$ pe cealaltă parte a lui H și $a = \inf h$.

După Zadeh, vom numi grad de separare prin H al grupărilor G_1 și G_2 numărul $b = 1 - a$.

Fie G_1 , G_2 și $G_k = G_1 \cap G_2$ grupări convexe mărginite, cu

$$M_1 = \sup \varphi_1(x),$$

$$M_2 = \sup \varphi_2(x),$$

$$M_k = \sup \varphi_k(x).$$

Atunci, $1 - M_k$ este cel mai mare grad de separare al grupărilor G_1 și G_2 , ce poate fi realizat cu un hiperplan în X .

Demonstrația acestei teoreme este dată în anexa 3.

Teorema de separare a grupărilor convexe poate fi utilizată pentru dimensionarea grupărilor, adică pentru determinarea pragului de selecție a elementelor într-o grupare și deci a suprapunerii grupărilor. Presupunem că o colecție de înregistrări trebuie împărțită în m grupări și că pentru aceasta se folosește una din funcțiile de apartenență analizate în paragrafele precedente împreună cu matricea probabilităților descriptorilor obținută prin metoda analizei claselor latente. Pentru fiecare grupare funcția de apartenență are parametri determinați de vectorii linie ai acestei matrice. Cu valorile pe care funcțiile de apartenență le iau pentru fiecare înregistrare se poate forma tabloul T_1 :

	G_1	G_2	. . .	G_m
x_1	$\varphi_1(x_1)$	$\varphi_2(x_1)$. . .	$\varphi_m(x_1)$
x_2	$\varphi_1(x_2)$	$\varphi_2(x_2)$. . .	$\varphi_m(x_2)$
.
.
x_r	$\varphi_1(x_r)$	$\varphi_2(x_r)$. . .	$\varphi_m(x_r)$

cu ajutorul căruia care se formează tabloul T'_2 :

	G_{12}	G_{13}		$G_{m-1,m}$
x_1	$\varphi_{12}(x_1)$	$\varphi_{13}(x_1)$	$\cdot \cdot \cdot$	$\varphi_{m-1,m}(x_1)$
x_2	$\varphi_{12}(x_2)$	$\varphi_{13}(x_2)$	$\cdot \cdot \cdot$	$\varphi_{m-1,m}(x_2)$
\cdot	\cdot	\cdot		\cdot
\cdot	\cdot	\cdot		\cdot
x_r	$\varphi_{12}(x_r)$	$\varphi_{13}(x_r)$	$\cdot \cdot \cdot$	$\varphi_{m-1,m}(x_r)$

Pentru suprapunerea a cel mult m grupări, pragul de selecție L trebuie ales astfel ca

$$L < \min_{ij} \max_h \varphi_{ij}(x^h).$$

În acest fel problema clasificării se reduce la efectuarea următoarelor etape succesive:

- calculul valorilor p'_k ,
- determinarea tablourilor T_1 și T_2 ,
- alegerea pragului de selecție după tabloul T_2 ,
- gruparea elementelor $x \in X$ comparînd valorile funcției de apartenență cu pragul de selecție.

O grupare i este definită deci de expresia

$$[\forall x \in G] (\varphi_i(x) \geq L).$$

Pentru exemplificare presupunem o colecție pentru care matricea probabilităților p'_k este

0,62754	0,68694	0,06197	0,57853
0,59984	0,13551	0,29430	0,51050
0,76266	0,27440	0,52651	0,61937
0,45522	0,32918	0,97940	0,75953

În tabela 3 sînt date valorile funcției

$$\frac{\sum_{k=1}^n p_k^i d_k(x)}{n_i},$$

unde n_i este numărul de descriptori nenuli într-o înregistrare pentru o colecție de 16 înregistrări cu 4 descriptori, care trebuie divizată în patru grupări.

Tabela 3

Înregistrare		G_1	G_2	G_3	G_4
x_1	0 0 0 0	0	0	0	0
x_2	1 0 0 0	0,62754	0,59984	0,76266	0,45522
x_3	0 1 0 0	0,68694	0,13551	0,27440	0,32918
x_4	0 0 1 0	0,06197	0,29430	0,52651	0,97949
x_5	0 0 0 1	0,57853	0,51050	0,61937	0,75953
x_6	1 1 0 0	0,65724	0,36767	0,51853	0,39220
x_7	1 0 1 0	0,34475	0,44707	0,64458	0,71731
x_8	1 0 0 1	0,60303	0,55517	0,69101	0,60737
x_9	0 1 1 0	0,37445	0,21490	0,40045	0,65429
x_{10}	0 1 0 1	0,63273	0,32300	0,44688	0,54435
x_{11}	0 0 1 1	0,32025	0,40240	0,57294	0,86946
x_{12}	1 1 1 0	0,45881	0,34321	0,52119	0,58793
x_{13}	1 1 0 1	0,63100	0,41528	0,55214	0,51464
x_{14}	1 0 1 1	0,42268	0,46821	0,63618	0,73138
x_{15}	0 1 1 1	0,44248	0,31344	0,47343	0,68937
x_{16}	1 1 1 1	0,48874	0,38504	0,54573	0,63083

Se observă că

$$[\forall x](\varphi_2(x) < \varphi_3(x))$$

și deci

$$G_2 \subset G_3,$$

adică această funcție conduce numai la 3 grupări.

În tabela 4 sînt date valorile funcțiilor corespunzătoare grupărilor intersecție.

Tabela 4

Înregistrare	G_{13}	G_{14}	G_{34}
x_2	0,62754	0,45522	0,45522
x_3	0,27440	0,32918	0,27440
x_4	0,06197	0,06197	0,52651
x_5	0,57853	0,57853	0,61937
x_6	0,51853	0,39220	0,39220
x_7	0,34475	0,34475	0,64458
x_8	0,60303	0,60303	0,60737
x_9	0,37445	0,37445	0,40045
x_{10}	0,44688	0,54435	0,44688
x_{11}	0,32025	0,32025	0,57294
x_{12}	0,45881	0,45881	0,52119
x_{13}	0,55214	0,51464	0,51464
x_{14}	0,42264	0,42268	0,63618
x_{15}	0,44248	0,44248	0,47343
x_{16}	0,48874	0,48874	0,54573

În această tabelă

$$\max \varphi_{13} = 0,62754,$$

$$\max \varphi_{14} = 0,60303,$$

$$\max \varphi_{34} = 0,64458.$$

În tabela 3 cea mai mare valoare mai mică decât 0,60303 în coloanele G_1, G_3, G_4 este 0,58793. Cu o valoare de prag 0,58793 se obțin grupările din tabela 5.

Tabela 5

G_1	x_2	x_3	x_6	x_8	x_{10}	x_{13}					
G_3	x_2	x_5	x_7	x_8	x_{14}						
G_4	x_4	x_5	x_7	x_8	x_9	x_{11}	x_{12}	x_{14}	x_{15}	x_{16}	

În tabela 6 sînt date valorile funcției Baker^{*)} pentru colecția din tabela 3.

*) După Winters [189].

Tabela 6

Înregistrări					G_1	G_2	G_3	G_4
x_1	0	0	0	0	0,7546	0,1695	0,0734	0,0023
x_2	1	0	0	0	0,6278	0,3024	0,0686	0,0010
x_3	0	1	0	0	0,3432	0,1860	0,4673	0,0033
x_4	0	0	1	0	0,5077	0,3041	0,0078	0,1802
x_5	0	0	0	1	0,6720	0,2356	0,0860	0,0063
x_6	1	1	0	0	0,2704	0,3142	0,4138	0,0014
x_7	1	0	1	0	0,4000	0,5138	0,0069	0,0791
x_8	1	0	0	1	0,5261	0,3954	0,0757	0,0027
x_9	0	1	1	0	0,2650	0,3831	0,0571	0,2946
x_{10}	0	1	0	1	0,2726	0,2306	0,4886	0,0080
x_{11}	0	0	1	1	0,3299	0,3084	0,0066	0,3548
x_{12}	1	1	1	0	0,2015	0,6246	0,0488	0,1249
x_{13}	1	1	0	1	0,2064	0,3742	0,4158	0,0033
x_{14}	1	0	1	1	0,2757	0,5526	0,0062	0,1653
x_{15}	0	1	1	1	0,1448	0,3265	0,0411	0,4875
x_{16}	1	1	1	1	0,1245	0,6020	0,0397	0,2336

În tabela 7 sînt date valorile funcțiilor care corespund grupărilor intersecție.

Tabela 7

	G_{12}	G_{13}	G_{14}	G_{23}	G_{24}	G_{34}
x_1	0,1695	0,0734	0,0023	0,0734	0,0023	0,0023
x_2	0,3024	0,0686	0,0010	0,0686	0,0010	0,0010
x_3	0,1860	0,3432	0,0033	0,1860	0,0033	0,0033
x_4	0,3041	0,0078	0,1802	0,0078	0,1802	0,0078
x_5	0,2356	0,0860	0,0063	0,0860	0,0063	0,0063
x_6	0,2704	0,2704	0,0014	0,3142	0,0014	0,0014
x_7	0,4000	0,0069	0,0791	0,0069	0,0791	0,0069
x_8	0,3954	0,0757	0,0027	0,0757	0,0027	0,0027
x_9	0,2650	0,0571	0,2650	0,0571	0,2946	0,0571
x_{10}	0,2306	0,2726	0,0080	0,2306	0,0080	0,0080
x_{11}	0,3084	0,0066	0,3299	0,0066	0,3084	0,0066
x_{12}	0,2015	0,2015	0,1249	0,0488	0,1249	0,0488
x_{13}	0,2064	0,2064	0,0033	0,3742	0,0033	0,0033
x_{14}	0,2757	0,0062	0,1653	0,0062	0,1653	0,0062
x_{15}	0,1448	0,0411	0,1448	0,0411	0,3265	0,0411
x_{16}	0,1245	0,0397	0,1245	0,0397	0,2336	0,0397

În această tabelă

$$\begin{aligned}\max \varphi_{12}(x) &= 0,4, \\ \max \varphi_{13}(x) &= 0,3432, \\ \max \varphi_{14}(x) &= 0,3299, \\ \max \varphi_{23}(x) &= 0,3742, \\ \max \varphi_{24}(x) &= 0,3265, \\ \max \varphi_{34}(x) &= 0,0571.\end{aligned}$$

Considerînd 0,0571 ca valoare de prag se obțin grupări cu suprapunere foarte mare, care aproape se confundă. În acest caz, deoarece $\max \varphi_4 > \max \varphi_{24}$ este convenabil să se aleagă ca prag valoarea imediat superioară, adică 0,3265.

În tabela 8 sînt date grupările determinate de acest prag.

Tabela 8

G_1	x_1	x_2	x_3	x_4	x_5	x_7	x_8	x_{11}
G_2	x_7	x_8	x_9	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
G_3	x_3	x_6	x_{10}	x_{13}				
G_4	x_{11}	x_{15}						

Cele două funcții cu care s-a făcut exemplificarea conduc la grupări diferite. Rezultatul procesului de selecție este însă același deoarece selecția se face corespunzător modului de clasificare. În fiecare caz sistemului i se prezintă o cerere de selecție care la rîndul ei este clasificată într-una sau mai multe grupări ale sistemului, ca oricare dintre înregistrări. Astfel o cerere de forma

$$q = \{1,0,1,0\}$$

va fi clasificată în primul caz la G_3 și G_4 și în al doilea caz la G_1 și G_2 , deoarece pentru aceste grupări $\varphi(q)$ depășește valoarea de prag. Din tabela A. 2 (din anexa 1) se vede că strategiile cu α_C α_{PN} α_{MM} π_{RS} conduc la interogarea înregistrărilor pe care sistemul le-ar fi interogată dacă nu se făcea clasificarea colecției.

5.5.5. Grupări compacte

Fie două funcții de apartenență φ_1 și φ_2 . Vom aplica regula de decizie

$$\frac{\varphi_1(x)}{\varphi_2(x)} > c \rightarrow x \in G_1.$$

Luînd relația de mai sus ca egalitate este definită o frontieră, adică o suprafață de separare a două submulțimi compacte. Punînd $c = 1$ înseamnă că aplicăm criteriul probabilității minime de eroare cînd frontiera corespunde conturului pe care cele două funcții sînt identice. În acest caz regula de decizie devine

$$\varphi_1(x) - \varphi_2(x) > 0 \rightarrow x \in G_1.$$

Cele m funcții $\varphi_1, \varphi_2, \dots, \varphi_m$ definesc deci m suprafețe de decizie care determină m regiuni în spațiul înregistrărilor.

Clasificarea unei înregistrări se reduce în acest caz la calcularea celor m funcții și repartizarea înregistrării la acea regiune care corespunde funcției cu valoare maximă.

Deoarece grupările obținute astfel sînt mai mici și cererea de selecție este clasificată numai la o singură grupare, numărul înregistrărilor selectate va fi mai mic decît în cazul cînd este interogată întreaga colecție.

În tabela 9 sînt date grupările obținute cu această regulă de decizie pentru funcția din tabela 3.

Tabela 9

G_1	x_3	x_6	x_{10}	x_{13}						
G_3	x_2	x_8								
G_4	x_4	x_5	x_7	x_9	x_{11}	x_{12}	x_{14}	x_{15}	x_{16}	

În tabela 10 sînt date grupările obținute pentru funcția din tabela 6.

Tabela 10

G_1	x_2	x_4	x_5	x_8	
G_2	x_7	x_9	x_{12}	x_{14}	x_{15}
G_3	x_3	x_6	x_{10}	x_{13}	
G_4	x_{11}	x_{16}			

Se poate trage concluzia că pentru sistemele de selecție existența grupărilor suprapuse permite sistemului să dea răspunsuri sensibile mai bune și că deci pentru aceste sisteme nu sînt indicate metodele de clasificare care conduc la grupări compacte.

5.6. SISTEME INTERACTIVE

Sistemele automate de regăsire a informațiilor fiind sisteme mecanice suferă de o inevitabilă inflexibilitate. Nevoile beneficiarilor unei colecții mari de documente sînt prea dispersate pentru a putea fi satisfăcute cu un singur algoritm oricît de atent elaborat.

O cale pentru a evita acest dezavantaj o constituie utilizarea informației de reacție de la beneficiar în timpul procesului de regăsire. Acest deziderat poate fi realizat astfel:

- beneficiarul prezintă o cerere de selecție;
- sistemul de regăsire furnizează unele informații privind un anumit număr de documente considerate relevante la cerere;
- din aceste documente beneficiarul selecționează pe acelea pe care el le consideră relevante la cererea sa și furnizează această informație sistemului;
- sistemul efectuează o altă regăsire ținînd seama de raționamentul beneficiarului.

Ultimele etape pot fi repetate de mai multe ori. Un astfel de proces este cunoscut sub numele de „reacție de relevanță”, reacție ce există ca urmare a interacțiunii în timp real dintre beneficiar și calculator.

Fie X_R submulțimea nevidă a înregistrărilor considerate relevante. Atunci o cerere optimală este o cerere care permite discriminarea maximă între submulțimea X_R și restul înregistrărilor.

Dacă $\bar{\gamma}(q, x)$ este media funcției de selecție folosită pentru a confrunța cererea q cu înregistrările unei colecții X , atunci o cerere optimă q_0 poate fi definită ca fiind cererea care maximizează funcția

$$\Gamma = \bar{\gamma}_{x \in X_R}(q, x) - \bar{\gamma}_{x \notin X_R}(q, x).$$

În practică expresia de mai sus nu este prea utilă și în loc să se determine direct valoarea q_0 se fac o serie de aproximări pornind de la cererea inițială care identifică o parte a submulțimii X_R .

Presupunem r înregistrări din care r_0 sînt identificate ca imagini ale unor documente relevante. Atunci funcția Γ pentru a fi maximizată trebuie scrisă ca

$$\Gamma = \frac{1}{r_0} \sum_{x \in X_R} \gamma(x, q) - \frac{1}{r - r_0} \sum_{x \notin X_R} \gamma(x, q).$$

Cînd γ este înlocuită cu funcția cosinus, se obține

$$\begin{aligned} \Gamma &= \frac{1}{r_0} \sum_{x \in X_R} \frac{qx}{|q||x|} - \frac{1}{r - r_0} \sum_{x \notin X_R} \frac{qx}{|q||x|} = \\ &= \frac{q}{|q|} \left[\frac{1}{r_0} \sum_{x \in X_R} \frac{x}{|x|} - \frac{1}{r - r_0} \sum_{x \notin X_R} \frac{x}{|x|} \right]. \end{aligned}$$

Expresia de mai sus este de forma unui produs $\Gamma = q \cdot a$ astfel încît cererea q_0 care maximizează Γ va fi proporțională cu a , adică

$$q_0 = k \left[\frac{1}{r_0} \sum_{x \in X_R} \frac{x}{|x|} - \frac{1}{r - r_0} \sum_{x \notin X_R} \frac{x}{|x|} \right].$$

Atunci algoritmul de modificare a cererii poate fi scris sub forma

$$q_{i+1} = n_r n_s q_i + n_r \sum_{i=1}^{n_r} \frac{y_i}{|y_i|} - n_s \sum_{i=1}^{n_s} \frac{z_i}{|z_i|},$$

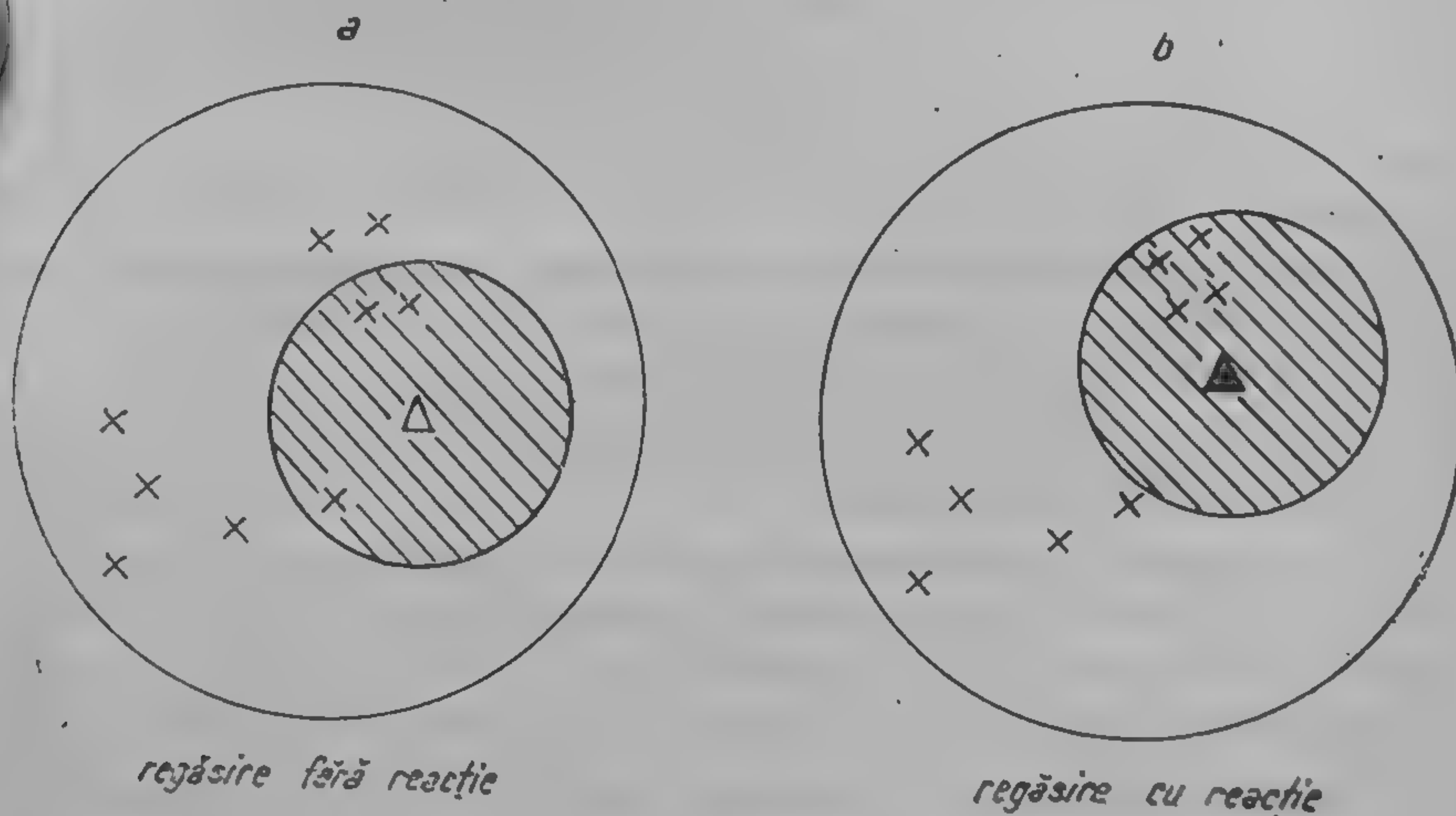
unde q_i este cererea i în secvență, $Y = \{y_1, y_2, \dots, y_{n_r}\}$ mulțimea vectorilor înregistrare corespunzători documentelor relevante regăsite ca răspuns la cererea q_i , $Z = \{z_1, z_2, \dots, z_{n_s}\}$ mulțimea vectorilor înregistrare corespunzători documentelor nerelevante regăsite ca răspuns la cererea q_i , iar n_r și n_s reprezintă numărul documentelor relevante, respectiv nerelevante, regăsite de beneficiar în etapa i .

Specificarea mulțimilor Y și Z constituie reacția de la beneficiar după etapa i a procesului.

Vectorilor înregistrare [și cerere le corespund puncte în spațiul înregistrărilor. Sistemul de regăsire selectează în acest spațiu toate înregistrările care se găsesc „aproape” de cerere (fig. 19, a)

Modificările datorite reacției de relevanță sînt folosite pentru a muta cererea într-o nouă poziție în spațiul înregistrărilor, acolo unde densitatea documentelor relevante este mai mare (fig. 19, b).

O extindere a procedurii standard de reacție de relevanță poate fi făcută prin metoda segmentării cererii (*query splitting*). Considerăm exemplul din figura 20. Dacă un algoritm simplu de reacție mută cererea originală q către un grup de înregistrări ce corespund unor documente relevante, atunci nu vor mai fi regăsite documentele relevante aparținînd altui grup. Pentru a evita aceste pierderi, cererea q se înlocuiește cu alte cereri (în cazul nostru cererile q_1 și q_2). Beneficiarul face atunci raționamente de relevanță



- X înregistrări corespunzînd documentelor relevante
- Δ cerere nemodificată
- ▲ cerere modificată

Fig. 19

asupra documentelor regăsite de q_1 și q_2 și algoritmul de reacție de relevanță se aplică fiecărei cereri q_1 și q_2 separat, obținându-se noi cereri q_3 și q_4 și așa mai departe.

În anumite cazuri o perfecționare a procesului de interacțiune

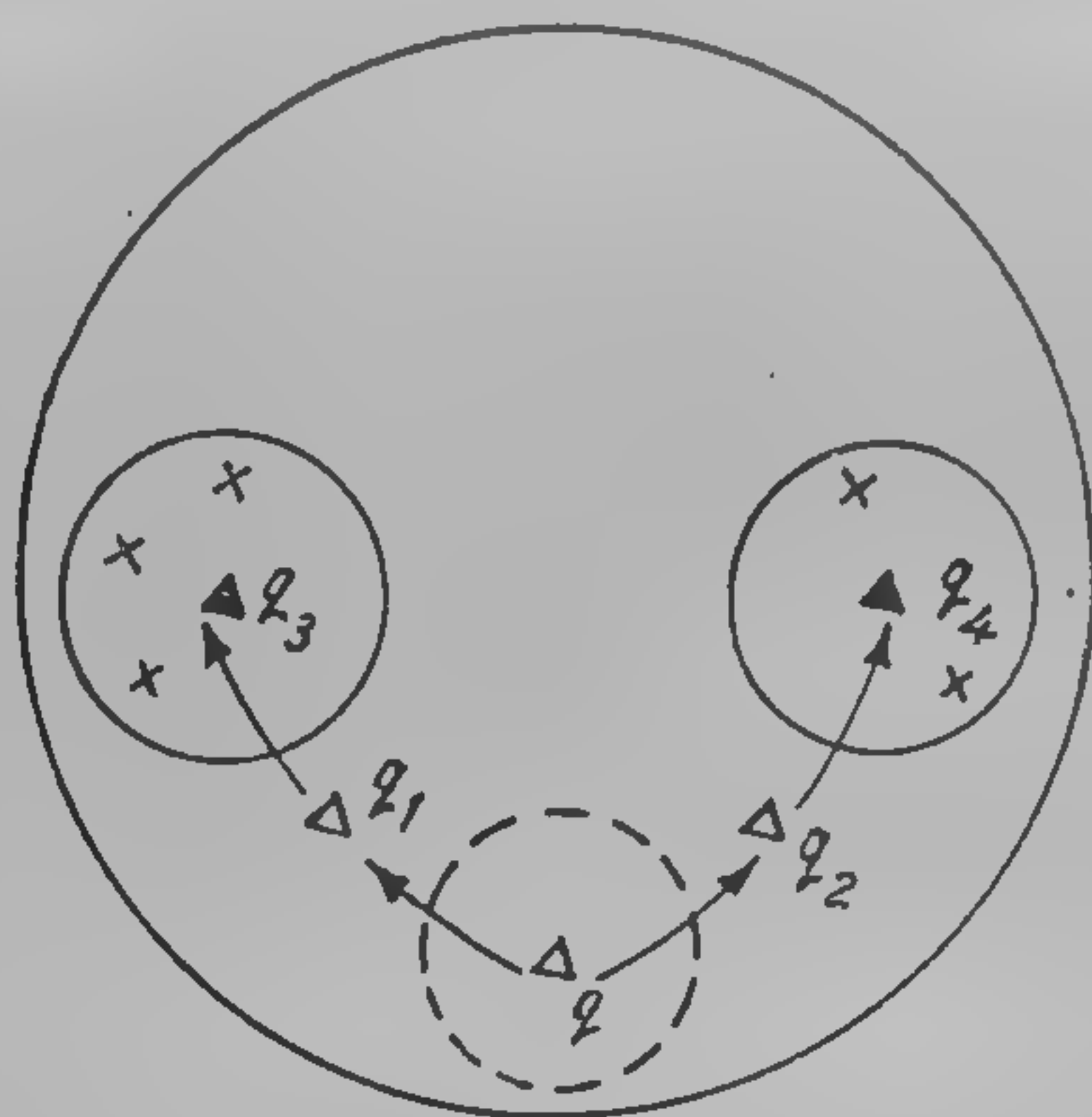


Fig. 20

poate fi realizată folosind drept cerere optimală chiar un vector înregistrare.

Această situație este recomandabilă când beneficiarul unui sistem de regăsire automată își cunoaște foarte bine domeniul și vine la sistem cu o idee foarte clară privind tipul de document pe care îl dorește. Astfel el poate stabili un document „foarte relevant” care este definit apoi drept cerere cu care se face selecția finală. Figura 21 ilustrează procesul.

Cu toate că modificările aduse cererii, așa cum au fost discutate până acum, rectifică într-o oarecare măsură imperfecțiunile cererii inițiale prezentată sistemului, ele nu afectează spațiul înregistrărilor. Orice imperfecțiune a acestui spațiu se păstrează atît timp cît există colecția. S-ar părea deci că pentru a se obține rezultate optime ar trebui modificat și spațiul înregistrărilor. Este evident că în sistemele cu clasificare automată, unde înregistrările sînt grupate în zone distanțate, nu este suficientă numai modificarea cererii, pentru că prin modificare cererea va fi mutată într-o poziție apropiată de o grupare și depărtată de alte grupări care în felul acesta vor fi ignorate. Presupunerea că documentele găsite ca relevante la

o cerere dată sînt de fapt interconectate conduce la concluzia că este necesară o nouă regrupare în spațiul înregistrărilor ținînd seama de datele reacției de relevanță. Numai în felul acesta determinarea

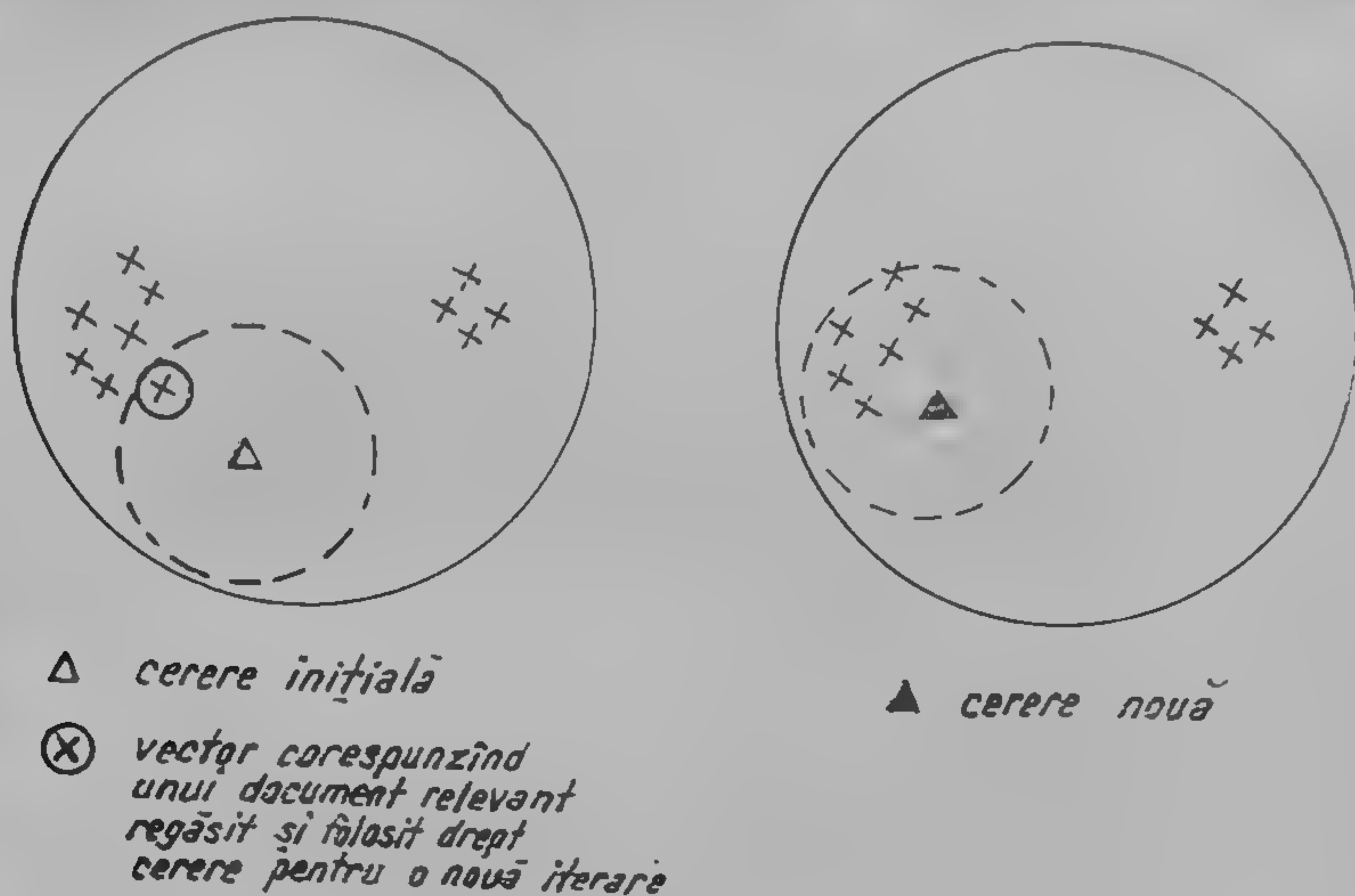


Fig. 21

unei înregistrări corespunzătoare unui document relevant va ușura găsirea celorlalte.

În figura 22, *a* este prezentat din nou spațiul din figura 19 în care poziția înregistrărilor este schimbată astfel ca la modificarea cererii (fig. 22, *b*) să fie regăsite aproape toate înregistrările ce corespund documentelor relevante.

Se ajunge astfel la noțiunea de „spațiu dinamic” pornind de la următoarele presupuneri:

— pentru o cerere dată, descriptorii care apar mai des în documentele relevante decît în documentele nerelevante contribuie probabil în mod substanțial la relevanța documentelor pertinente. Descriptorii semnificativi sînt legați între ei și apar adesea simultan. Mărind valoarea acestora se realizează o apropiere a documentelor similare.

— orice document relevant (determinat prin reacție) care nu conține un descriptor semnificativ conține probabil informații legate de acesta și deci acest descriptor trebuie adăugat documentului.

Este dificil să se determine un criteriu pentru se decide asupra semnificației descriptorilor pentru un anumit document. Poate fi calculat însă un factor de discriminare δ_i , folosind mărimile r_i și

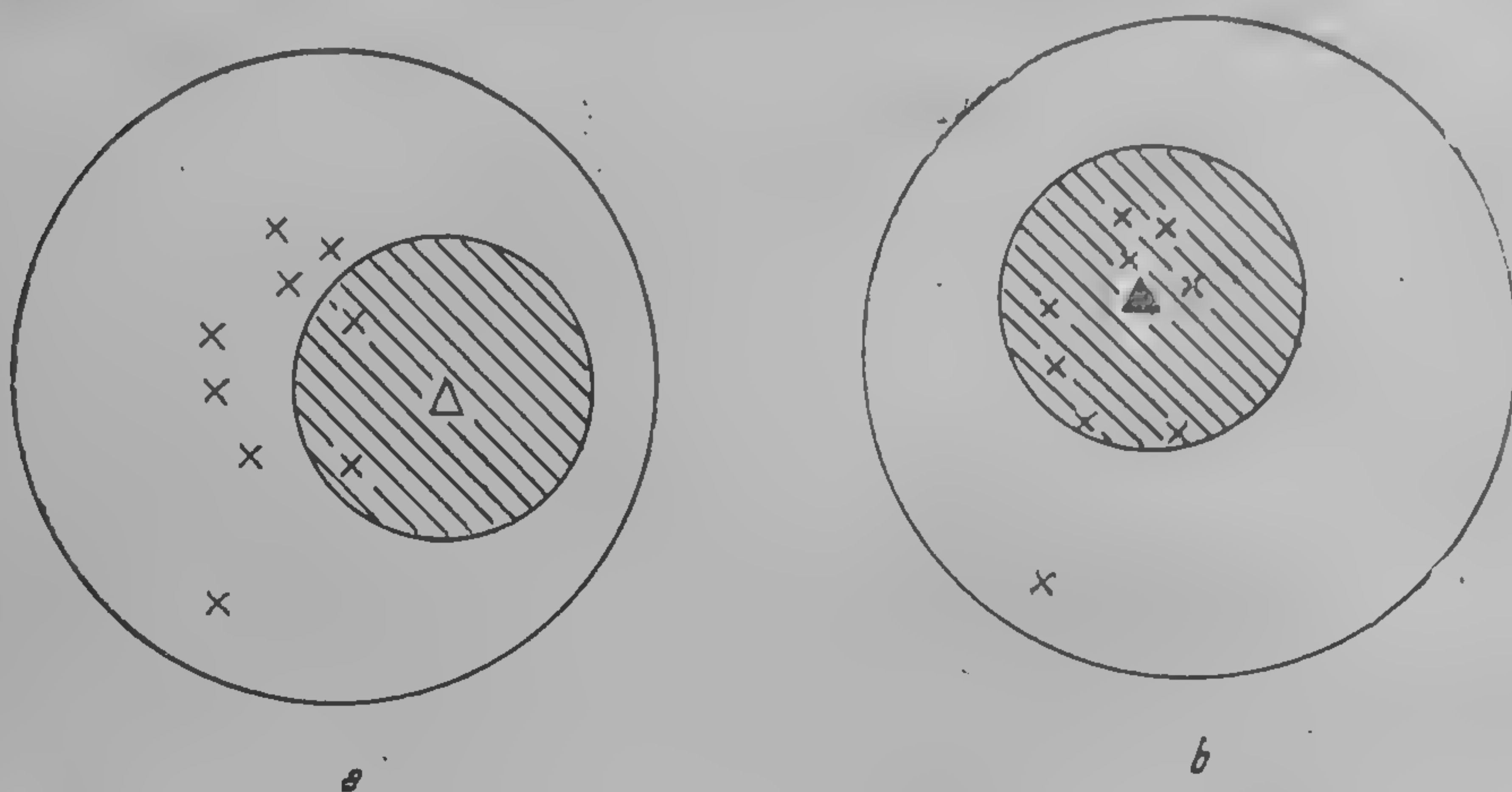


Fig. 22

n_i , unde r_i , n_i , și δ_i sînt

$$r_i = \frac{1}{I} \sum_{k \in R} v_{ki},$$

$$n_i = \frac{1}{J} \sum_{k \in N} v_{ki},$$

$$\delta_i = \frac{r_i - n_i}{r_i + n_i}.$$

în care

- I — numărul de elemente al mulțimii R ,
- J — numărul de elemente al mulțimii N ,
- R — mulțimea elementelor relevante regăsite,
- N — mulțimea elementelor nerelevante regăsite,
- v_{ki} — valoarea descriptorului i în documentul k ,
- r_i — valoarea medie a descriptorului i în documentele relevante regăsite,

n_i — valoarea medie a descriptorului i în documentele nerelevante regăsite.

Diferența $r_i - n_i$, dacă este pozitivă, este o măsură indicînd cu cît este mai important descriptorul i în descrierea naturii documentelor relevante decît în descrierea naturii documentelor nerelevante. Această măsură, cînd este normalizată prin împărțire cu factorul $r_i + n_i$, devine factorul discriminant δ_i . O valoare pozitivă pentru δ_i indică deci că descriptorul apare mai frecvent în documentele relevante regăsite decît în documentele nerelevante regăsite și, în concluzie, are o anumită semnificație. Cu cît valoarea δ_i este mai mare, cu atît mai semnificativ este descriptorul ca indicator al relevanței documentului.

Toate înregistrările în spațiul înregistrărilor sînt apoi modificate după expresia

$$v'_{ki} = v_{ki} (1 + k \delta_i)$$

pentru un k ales în mod corespunzător.

Este posibil ca modificarea spațiului înregistrărilor să se facă operînd numai asupra vectorilor ce corespund documentelor relevante și nu asupra tuturor vectorilor ca mai sus.

În acest caz vectorul cerere q_0 este modificat de cîteva ori folosind o tehnică standard de reacție de relevanță pînă cînd se obține un vector q_n care regăsește o mulțime de înregistrări X_D acceptabilă pentru beneficiar. Valorile descriptorilor în vectorii ce aparțin mulțimii X_D sînt apoi modificate pentru a micșora unghiul dintre fiecare vector $x_i \in X_D$ și vectorul q_0 . O descreștere a unghiului dintre vectorii q_0 și x_i înseamnă o creștere a cosinusului:

$$\cos(q_0, x_i) = \frac{\sum_{j=1}^n q_0^j x_i^j}{\left(\sum_{j=1}^n (q_0^j)^2 \sum_{j=1}^n (x_i^j)^2 \right)^{1/2}},$$

unde x_i^j este valoarea descriptorului j în vectorul x_i .

Pentru vectorul cerere q_0 și înregistrările $x_i \in X_D$, procesul de modificare are loc în două etape. Vectorul cerere este mai întîi normalizat la „lungimea” vectorului înregistrare făcînd suma valorilor componentelor lui q_0 egală cu suma corespunzătoare pentru x_i . Dacă A_0 este suma valorilor componentelor vectorului cerere și A_i este suma pentru vectorul înregistrare, atunci vectorul cerere nor-

malizat este definit ca

$$\bar{q}_0 = q_0 \frac{A_d}{A_q}.$$

Valorile vectorului înregistrare corespunzător unui document relevant x_i sînt modificate atunci după expresia

$$\bar{x}_i^j = x_i^j + \lambda (\bar{q}_0^j - x_i^j), \quad 0 < \lambda < 1.$$

Vectorul cerere fiind normalizat, modificarea de mai sus a presupus vectori de lungime egală. Ea este deci o transformare liniară, adică \bar{x}_i va avea aceeași lungime ca și x_i . Lungimea vectorului x_i este $\sum_j x_i^j$, deci

$$\sum_j \bar{x}_i^j = \sum_j x_i^j + \lambda \left(\sum_j \bar{q}_0^j - \sum_j x_i^j \right).$$

Deoarece

$$\sum_j \bar{q}_0^j = \sum_j x_i^j,$$

atunci

$$\sum_j \bar{x}_i^j = \sum_j x_i^j.$$

5.7. OBSERVAȚII BIBLIOGRAFICE

Majoritatea metodelor de clasificare cunoscute se bazează pe matricea de similitudine dintre înregistrări.

Similitudinea dintre documente ținînd seama de descriptorii a fost analizată de Parker-Rhodes [136], Salton [158], Bonner [15] și Sanders [166]. O altă modalitate de a construi matricea de similitudine dintre documente o constituie folosirea referințelor bibliografice. Salton [154], Garfield [14] și Chien și Preparata [21] au studiat această modalitate considerînd mulțimea X a documentelor ca avînd

structură de graf neorientat. Nodurile acestui graf reprezintă elementele mulțimii X , iar laturile reprezintă cuplajele bibliografice w_{ab} :

$w_{ab} = 1$ dacă documentul a citează pe b sau dacă documentul a este citat de b ;

$w_{ab} = 0$ dacă documentul a nu citează sau nu este citat de b .

Kessler [74] Price și Schiminovich [139] au definit intensitatea de cuplaj ca numărul de referințe comune la două documente.

O sinteză a metodelor de clasificare pe baza matricei de similitudine elaborate la IBM a fost făcută de Bonner [15].

Parker-Rhodes [137] și Needham [103], [104], ambii de la Cambridge Language Research Unit, au elaborat teoria matematică a grupărilor bazată pe noțiunea de predispoziție. Metoda vectorilor proprii pentru găsirea grupărilor inițiale se datorește lui LeSchack [81] iar cazul matricei de similitudine fără elemente nule este prezentat după metoda indicată de Ghelfand [45].

Sparck-Jones și Jackson [177], tot de la Cambridge Language Research Unit, au folosit alte definiții ale grupării care nu țin seama de similitudinea fiecărui membru al grupării, ci de proprietățile partiției. Ei au introdus noțiunea de coeziune la frontieră. Coeziunea la o frontieră, care împarte mulțimea în două submulțimi A și B , este

$$\frac{s_{AB}}{s_{AA} + s_{BB}} \text{ sau } \frac{s_{AB}^2}{s_{AA} + s_{BB}}$$

unde s_{AB} este similitudinea dintre mulțimile A și B ,

$$s_{AB} = \sum s_{a_i b_j},$$

a_i fiind elementele pe o parte a frontierei și b_j elementele pe cealaltă parte a frontierei.

Dacă $s_{AB} = 0$, atunci A și B sînt disjuncte. Valoarea s_{AB} indică gradul de suprapunere. Raportul s_{AB}/s_{AA} măsoară izolarea grupării. Cu cît este mai mică valoarea acestui raport cu atît gruparea este mai izolată.

Dacă f_A este numărul de elemente în gruparea A atunci $(f_A - 1) f_A$ măsoară limita superioară a lui s_{AA} și de aceea raportul $(f_A - 1) f_A / s_{AA}$ măsoară „plinătatea” lui s_{AA} . Diferite combinații ale acestor factori au fost încercate pentru definirea unei funcții de coeziune ca de exemplu

$$\frac{s_{AB}}{s_{AA}} \cdot \frac{f_A^2 - 1}{s_{AA}}.$$

Primul factor dă o imagine externă a grupării, iar cel de al doilea factor dă o imagine internă. De observat că în acest caz s_{BB} nu mai apare.

Parker-Rhodes [136] a demonstrat că definițiile grupării bazate pe coeziune sînt echivalente cu cele precedente.

Mulți cercetători au investigat utilizarea grafelor pentru a efectua clasificarea.

Abraham [1], [2] la IBM a utilizat un procedeu de clasificare bazat pe arbori. O metodă originală este utilizată de Paul Constantinescu [25].

Preparata și Chien [138] transformă graful într-o rețea unidimensională la care pentru fiecare locație se calculează o funcție. Ei dau un algoritm pentru modificarea funcției prin reorganizarea rețelei, ceea ce conduce la obținerea de grupări suprapuse. Acest algoritm este prezentat în paragraful 5.3.

Hill [61] a sugerat o metodă vectorială de clasificare. Conform acestei metode primul document este stabilit drept gruparea 1. A doua înregistrare este comparată cu gruparea 1 și dacă este similară este fixată de asemenea la gruparea 1; dacă nu este stabilită ca gruparea distinctă etc. Fiecare clasă este un vector linie al unei matrice $P = (p_{ij})$, unde $p_{ij} = n_{ij}/n_i$, n_{ij} fiind numărul de înregistrări în gruparea i cu descriptorul j , iar n_i fiind numărul total de înregistrări în gruparea i . Elementele matricei P sînt numere între 0 și 1 reprezentînd frecvența relativă cu care descriptorul corespunzător a fost folosit pentru a descrie înregistrările fixate la grupare. Ca măsură a similarității se utilizează produsul matricial XP , unde X este vectorul înregistrare. O înregistrare este fixată la gruparea pentru care factorul de similaritate are valoarea cea mai mare.

Rocchio [147] într-o teză de doctorat susținută la Harvard University propune o metodă care permite ca numărul de grupări să poată fi controlat și la fel volumul unei grupări și suprapunerea dintre grupări.

În spațiul înregistrărilor fiecare înregistrare este supusă unui test de densitate regională pentru a se determina dacă în vecinătate se găsesc alte înregistrări în număr suficient. Acest test pretinde ca un anumit număr de înregistrări să aibă o similaritate cu înregistrarea de probă peste o valoare de prag.

Înregistrările care cad la proba de densitate sînt considerate pierdute și nu mai sînt alese ca centre potențiale ale unor grupări. Dacă o înregistrare trece proba de densitate se alege o valoare de

prag, în funcție de numărul maxim și minim al elementelor în grupare și toate înregistrările a căror similaritate cu înregistrarea centrală au valori mai mari decât valoarea de prag definesc o grupare.

O excelentă tratare a clasificării cu ajutorul funcțiilor discriminante este cea a lui Nillson [130]. Clasificarea cu matrice instruibile se datorește lui Agamalova și Agopian [3].

Metoda analizei claselor latente a fost utilizată prima oară de Lazarsfeld [78] în studii de sociologie pentru determinarea atitudinii personalului din armată față de diverși factori. Baker [9] a sugerat posibilitatea aplicării metodei la sistemele de regăsire a documentelor.

Metode matriciale pentru obținerea parametrilor latenți au fost elaborate de Anderson [5], Gibson [47], Mandansky [92] și Winters [189].

Noțiunea de funcție de apartenență a fost introdusă de Zadeh [190].

Folosirea teoremei de separare a mulțimilor fuzzy în clasificarea automată pentru sisteme de regăsire a informațiilor a fost propusă în [123], [124], [125].

Rocchio [147], Salton [161], [162], Riddle, Horwitz, Dietz [145,] Brauen, Holt, Wilcox [20] și Ide [67] s-au ocupat de reacția de relevanță. Metoda segmentării cererii a fost propusă de Borodin, Kerr și Lewis [19] iar noțiunea de spațiu dinamic se datorește lui Davis, Linsky și Zelkowitz [33].

Crawford și Melzer [29] au propus folosirea documentelor relevante în locul cererii de selecție în procesul de regăsire.

ANEXA 1

EFICACITATEA STRATEGIILOR DE SELECȚIE CU FUNCȚII DE APROPIERE

Pentru a ilustra practic criteriul de apreciere a eficacității strategiilor, expus în §2.7, în tabela A.1 sînt prezentate valorile pe care le iau cîteva funcții de selecție pentru o colecție cu 16 înregistrări și 4 descriptori binari.

Tabela A.1

q	1 0 1 0	π_{RS}	α_{SM}	α_{MK}	α_C	α_{PRN}	α_{MM}	α_S
x_1	0 0 0 0	0	0	0	0	0	0	0
x_2	1 0 0 0	1/4	1	1	1/√2	1/2	1/2	1
x_3	0 1 0 0	0	0	0	0	0	0	0
x_4	0 0 1 0	1/4	1	1	1/√2	1/2	1/2	1
x_5	0 0 0 1	0	0	0	0	0	0	0
x_6	1 1 0 0	1/4	1/2	1/2	1/√4	1/3	1/3	1/2
x_7	1 0 1 0	2/4	1	1	1	1	1	1
x_8	1 0 0 1	1/4	1/2	1/2	1/√4	1/3	1/3	1/2
x_9	0 1 1 0	1/4	1/2	1/2	1/√4	1/3	1/3	1/2
x_{10}	0 1 0 1	0	0	0	0	0	0	0
x_{11}	0 0 1 1	1/4	1/2	1/2	1/√4	1/3	1/3	1/2
x_{12}	1 1 1 0	2/4	2/3	1	2/√6	2/3	2/3	1
x_{13}	1 1 0 1	1/4	1/3	0	1/√6	1/4	1/4	1/2
x_{14}	1 0 1 1	2/4	2/3	1	2/√6	2/3	2/3	1
x_{15}	0 1 1 1	1/4	1/3	0	1/√6	1/4	1/4	1/2
x_{16}	1 1 1 1	2/4	2/4	1/2	2/√8	2/4	2/4	1

În tabela A.2 sînt prezentate răspunsurile sistemului pentru strategiile realizate cu cererea de selecție și funcțiile din tabela A.1.

Tabela A.2

Funcția	Răspunsul sistemului
π_{RS}	$x_7 \ x_{12} \ x_{14} \ x_{16}$ $x_2 \ x_4 \ x_6 \ x_8 \ x_9 \ x_{11} \ x_{13} \ x_{15}$ $x_1 \ x_3 \ x_5 \ x_{10}$
α_{SM}	$x_1 \ x_4 \ x_7$ $x_{12} \ x_{14} \ x_{16}$ $x_6 \ x_8 \ x_9 \ x_{11}$ $x_{13} \ x_{15}$ $x_1 \ x_3 \ x_5 \ x_{10}$
α_{MK}	$x_2 \ x_4 \ x_7 \ x_{12} \ x_{14}$ $x_6 \ x_8 \ x_9 \ x_{11} \ x_{16}$ $x_1 \ x_3 \ x_5 \ x_{10} \ x_{13} \ x_{15}$
α_S	$x_2 \ x_4 \ x_7 \ x_{12} \ x_{14} \ x_{16}$ $x_6 \ x_8 \ x_9 \ x_{11} \ x_{13} \ x_{15}$ $x_1 \ x_3 \ x_5 \ x_{10}$
α_C α_{PRN} α_{MM}	x_7 $x_{12} \ x_{14}$ $x_2 \ x_4 \ x_{16}$ $x_6 \ x_8 \ x_9 \ x_{11}$ $x_{13} \ x_{15}$ $x_1 \ x_3 \ x_5 \ x_{10}$

Strategiile de selecție produc o partiție a mulțimii $X = \{x_1, \dots, x_{16}\}$ în submulțimile disjuncte

$$X_k = \{x \mid \alpha(x) = c_k\}.$$

Din tabela A.2 se observă că strategiile cu funcțiile α_C , α_{PRN} realizează o partiție cu mai multe submulțimi disjuncte, deci au o putere de selecție mai mare. Deoarece strategiile de selecție cu funcțiile α_S , α_{MK} , π_{RS} produc răspunsuri foarte apropiate, se poate trage concluzia că funcțiile respective au practic același număr de componente. Pentru același motiv se poate spune că funcția α_{MM} are același număr de componente cu funcțiile α_C și α_{PRN} . Această afirmație este adevărată însă numai pentru cazul vectorilor binari.

O examinare a tabelii A.1 permite verificarea imediată a criteriului de eficiență a strategiilor de selecție. Strategiile cu funcțiile π_{RS} , α_S , și α_{SM} dau același rang înregistrărilor x_{14} și x_{16} , adică nu sînt sensibile la prezența descriptorului d_2 . Or, este evident că

înregistrarea x_{14} este mai apropiată de cererea de selecție decât înregistrarea x_{16} .

În tabela A.3 sînt calculate valorile funcțiilor pentru aceeași cerere de selecție, colecția fiind modificată însă prin lungirea înregistrărilor. Tuturor înregistrărilor li s-au adăugat doi descriptori care nu figurează în cererea de selecție.

Tabela A.3

q	1 0 1 0 0 0	π_{RS}	α_S	α_{SM}	α_{MK}	α_C	α_{PRN}	α_{MM}
x_1	0 0 0 0 1 1	0	0	0	0	0	0	0
x_2	1 0 0 0 1 1	1/6	1/2	1/3	1/2	$1/\sqrt{5}$	1/4	1/4
x_3	0 1 0 0 1 1	0	0	0	0	0	0	0
x_4	0 0 1 0 1 1	1/6	1/2	1/3	1	$1/\sqrt{5}$	1/4	1/4
x_5	0 0 0 1 1 1	0	0	0	0	0	0	0
x_6	1 1 0 0 1 1	1/6	1/2	1/4	1/4	$1/\sqrt{8}$	1/5	1/5
x_7	1 0 1 0 1 1	2/6	1	2/4	1	$2/\sqrt{8}$	2/4	1/2
x_8	1 0 0 1 1 1	1/6	1/2	1/4	1/4	$1/\sqrt{8}$	1/5	1/5
x_9	0 1 1 0 1 1	1/6	1/2	1/4	1/4	$1/\sqrt{8}$	1/5	1/5
x_{10}	0 1 0 1 1 1	0	0	0	0	0	0	0
x_{11}	0 0 1 1 1 1	1/6	1/2	1/4	1/2	$1/\sqrt{8}$	1/5	1/5
x_{12}	1 1 1 0 1 1	2/6	1	2/5	1	$2/\sqrt{10}$	2/5	2/5
x_{13}	1 1 0 1 1 1	1/6	1/2	1/5	0	$1/\sqrt{10}$	1/6	1/6
x_{14}	1 0 1 1 1 1	2/6	1	2/5	1	$2/\sqrt{10}$	2/5	2/5
x_{15}	0 1 1 1 1 1	1/6	1/2	1/5	0	$1/\sqrt{10}$	1/6	1/6
x_{16}	1 1 1 1 1 1	2/6	1	2/6	1/2	$2/\sqrt{12}$	2/6	2/6

În tabela A.4 sînt date răspunsurile sistemului pentru colecția din tabela A.3 și strategiile din tabela A.1.

În tabela A.5 sînt date valorile funcțiilor pentru colecția de înregistrări din tabela A.2 lungimea înregistrărilor fiind mărită neuniform.

În tabela A.6 sînt date răspunsurile sistemului pentru strategiile din tabela A.5.

Se observă că față de situația precedentă răspunsurile sînt practic neschimbate.

În tabela A.7 sînt date valorile funcțiilor pentru colecția de înregistrări din tabela A.5, cererea de selecție fiind lungită.

În tabela A.8 sînt date răspunsurile sistemului pentru strategiile din tabela A.7.

Tabela A.4

Funcția	Răspunsul sistemului
π_{RS} α_S	x_7 x_{12} x_{14} x_{16} x_2 x_4 x_6 x_8 x_9 x_{11} x_{13} x_{15} x_1 x_3 x_5 x_{10}
α_{SM}	x_7 x_{12} x_{14} x_2 x_4 x_{16} x_6 x_8 x_9 x_{11} x_{13} x_{15} x_1 x_3 x_5 x_{10}
α_{MK}	x_4 x_7 x_{12} x_{14} x_2 x_{11} x_{16} x_6 x_8 x_9 x_1 x_3 x_5 x_{10} x_{13} x_{15}
α_C α_{PRN} α_{MM}	x_7 x_{12} x_{14} x_{16} x_2 x_4 x_6 x_8 x_9 x_{11} x_{13} x_{15} x_1 x_3 x_5 x_{10}

Tabela A.5

q	1 0 1 0 0 0 0 0 0 0	π_{RS}	α_S	α_{SM}	α_C	α_{PRN}
x_1	0 0 0 0 1 1 0 0 0 0	0	0	0	0	0
x_2	1 0 0 0 1 1 1 0 0 0	1/10	1/2	1/4	$1/\sqrt{8}$	1/5
x_3	0 1 0 0 1 1 0 1 0 0	0	0	0	0	0
x_4	0 0 1 0 1 1 0 0 1 0	1/10	1/2	1/4	$1/\sqrt{8}$	1/5
x_5	0 0 0 1 1 1 0 0 0 1	0	0	0	0	0
x_6	1 1 0 0 1 1 1 1 0 0	1/10	1/2	1/6	$1/\sqrt{12}$	1/7
x_7	1 0 1 0 1 1 1 0 1 0	2/10	1	2/6	$2/\sqrt{12}$	2/6
x_8	1 0 0 1 1 1 1 0 0 1	1/10	1/2	1/6	$1/\sqrt{12}$	1/7
x_9	0 1 1 0 1 1 0 1 1 0	1/10	1/2	1/6	$1/\sqrt{12}$	1/7
x_{10}	0 1 0 1 1 1 0 1 0 1	0	0	0	0	0
x_{11}	0 0 1 1 1 1 0 0 1 1	1/10	1/2	1/6	$1/\sqrt{12}$	1/7
x_{12}	1 1 1 0 1 1 1 1 1 0	2/10	1	2/8	$2/\sqrt{16}$	2/8
x_{13}	1 1 0 1 1 1 1 1 0 1	1/10	1/2	1/8	$1/\sqrt{16}$	1/9
x_{14}	1 0 1 1 1 1 1 0 1 1	2/10	1	2/8	$2/\sqrt{16}$	2/8
x_{15}	0 1 1 1 1 1 0 1 1 1	1/10	1/2	1/8	$1/\sqrt{16}$	1/9
x_{16}	1 1 1 1 1 1 1 1 1 1	2/10	1	2/10	$2/\sqrt{20}$	2/10

Tabela A.6

Funcția	Răspunsul sistemului
π_{RS} α_S	x_7 x_{12} x_{14} x_{16} x_2 x_4 x_6 x_8 x_9 x_{11} x_{13} x_{15} x_1 x_3 x_5 x_{10}
α_{SM}	x_7 x_2 x_4 x_{12} x_{14} x_{16} x_6 x_8 x_9 x_{11} x_{13} x_{15} x_1 x_3 x_5 x_{10}
α_C α_{PRN}	x_7 x_{12} x_{14} x_2 x_4 x_{16} x_6 x_8 x_9 x_{11} x_{13} x_{15} x_1 x_3 x_5 x_{10}

Tabela A.7

q	1 0 1 0 1 0 1 0 0 0	π_{RS}	α_S	α_{SM}	α_C	α_{PRN}
x_1	0 0 0 0 1 1 0 0 0 0	1/10	1/2	1/2	$1/\sqrt{8}$	1/5
x_2	1 0 0 0 1 1 1 0 0 0	3/10	3/4	3/4	$3/\sqrt{16}$	3/5
x_3	0 1 0 0 1 1 0 1 0 0	1/10	1/4	1/4	$1/\sqrt{16}$	1/7
x_4	0 0 1 0 1 1 0 0 1 0	2/10	2/4	2/4	$2/\sqrt{16}$	2/6
x_5	0 0 0 1 1 1 0 0 0 1	1/10	1/4	1/4	$1/\sqrt{16}$	1/7
x_6	1 1 0 0 1 1 1 1 0 0	3/10	3/4	3/6	$3/\sqrt{24}$	3/7
x_7	1 0 1 0 1 1 1 0 1 0	4/10	4/4	4/6	$4/\sqrt{24}$	4/6
x_8	1 0 0 1 1 1 1 0 0 1	3/10	3/4	3/6	$3/\sqrt{24}$	3/7
x_9	0 1 1 0 1 1 0 1 1 0	2/10	2/4	2/6	$2/\sqrt{24}$	2/8
x_{10}	0 1 0 1 1 1 0 1 0 1	1/10	1/4	1/6	$1/\sqrt{24}$	1/9
x_{11}	0 0 1 1 1 1 0 0 1 1	2/10	2/4	2/6	$2/\sqrt{24}$	2/8
x_{12}	1 1 1 0 1 1 1 1 1 0	4/10	4/4	4/8	$4/\sqrt{32}$	4/8
x_{13}	1 1 0 1 1 1 1 1 0 1	3/10	3/4	3/8	$3/\sqrt{32}$	3/9
x_{14}	1 0 1 1 1 1 1 0 1 1	4/10	4/4	4/8	$4/\sqrt{32}$	4/8
x_{15}	0 1 1 1 1 1 0 1 1 1	2/10	2/4	3/8	$2/\sqrt{32}$	2/10
x_{16}	1 1 1 1 1 1 1 1 1 1	4/10	4/4	4/10	$4/\sqrt{40}$	4/10

Tabela A.8

π_{RS}	α_S	α_{SM}	α_O, α_{PRN}
x_1, x_2, x_3, x_4	x_7, x_9, x_{11}, x_{14}	x_2	x_7
x_2, x_4, x_8, x_{14}	x_2, x_4, x_8, x_{13}	x_7	x_2
x_2, x_4, x_{11}, x_{13}	$x_1, x_3, x_9, x_{11}, x_{13}$	$x_1, x_3, x_9, x_{10}, x_{12}, x_{14}$	x_{10}, x_{14}
x_1, x_3, x_8, x_{10}	x_3, x_3, x_{10}	x_{10}	x_{10}
		x_{13}	x_8, x_9
		x_9, x_{11}	x_3, x_{13}
		x_3, x_8, x_{13}	x_1, x_{15}
		x_{10}	x_9, x_8
			x_{10}

Se observă că puterea de selecție a strategiilor a crescut indiferent de funcția selecție folosită. Se verifică deci și faptul că numărul dihotomiilor variază cu numărul componentelor funcțiilor de selecție și cu lungimea cererii de selecție. Cu cât cererea de selecție are mai mulți descriptori, cu atât mai puține componente sînt anulate.

În tabela 9 se dau valorile unor funcții de selecție pentru o colecție de înregistrări ale căror componente iau valorile 1, 2, 3.

Tabela A.9

q	1 0 1 0 0 1 0 0 0	π_{RS}	α_S	α_{SM}	α_O	α_{MM}	α_{PRN}
x_1	1 2 3 0 0 1 0 0 0	5/9	1	5/7	$5/\sqrt{45}$	3/7	5/13
x_2	1 0 0 0 2 3 1 0 0	4/9	2/3	4/7	$4/\sqrt{45}$	2/8	4/14
x_3	3 1 2 0 0 1 0 0 0	6/9	1	6/7	$6/\sqrt{45}$	3/7	6/12
x_4	1 1 0 0 3 0 2 0 0	1/9	1/3	1/7	$1/\sqrt{45}$	1/9	1/17
x_5	0 3 0 0 0 2 2 0 0	2/9	1/3	2/7	$2/\sqrt{45}$	1/9	2/16
x_6	0 1 0 0 0 0 1 0 1	0	0	0	0	0	0
x_7	1 1 0 1 1 2 0 1 3	3/9	1/3	3/7	$3/\sqrt{45}$	2/11	3/15
x_8	1 1 0 2 1 0 2 2 0	2/9	1/3	2/9	$2/\sqrt{45}$	1/13	2/16
x_9	2 1 2 0 1 2 1 0 0	6/9	1	6/9	$6/\sqrt{45}$	3/9	6/12
x_{10}	0 3 0 0 0 1 0 0 0	1/9	1/3	1/4	$1/\sqrt{30}$	1/6	1/12
x_{11}	1 0 1 0 0 1 0 0 0	3/9	1	1	1	1	1
x_{12}	1 3 1 0 0 1 0 0 0	3/9	1	3/6	$3/\sqrt{36}$	3/6	3/12
x_{13}	1 1 1 0 0 1 0 0 0	3/9	1	3/4	$3/\sqrt{12}$	3/4	3/4
x_{14}	1 2 1 0 0 1 0 0 0	3/9	1	3/5	$3/\sqrt{21}$	3/5	3/7
x_{15}	1 2 1 1 0 1 0 0 0	3/9	1	3/6	$3/\sqrt{24}$	3/6	3/8
x_{16}	1 2 1 2 0 1 0 0 0	3/9	1	3/7	$3/\sqrt{33}$	3/7	3/11
x_{17}	1 2 1 3 0 1 0 0 0	3/9	1	3/8	$3/\sqrt{48}$	3/8	3/16

În tabela A.10 sînt prezentate răspunsurile sistemului pentru strategiile din tabela A.9.

Tabela A.10

α_{SM}	α_C	α_{PRN}	α_{MM}
x_{11} x_3 x_{13} x_1 x_9 x_{14} x_2 $x_{12}x_{15}$ x_7x_{16} x_{17} x_5 x_{10} x_8 x_4 x_6	x_{11} x_{13} x_3x_9 x_{14} x_1 x_{15} x_2 x_{16} x_{12} x_7 x_{17} x_5x_8 x_{10} x_4 x_6	x_{11} x_{13} x_3x_9 x_{14} x_1 x_{15} x_2 x_{16} x_{12} x_7 x_{17} x_5x_8 x_{10} x_4 x_6	x_{11} x_{13} x_{14} $x_{12}x_{15}$ $x_1x_3x_{16}$ x_{17} x_9 x_2 x_7 x_4x_5 x_8 x_{10} x_6
α_S	$x_1x_3x_{11}x_{12}x_{13}x_{14}x_{15}x_{16}x_{17}$ x_2 $x_4x_5x_7x_8x_{10}$ x_6		
π_{RS}	x_3x_9 x_1 x_2 $x_7x_{11}x_{12}x_{13}x_{14}x_{15}x_{16}x_{17}$ x_5x_8 x_4x_{10} x_6		

Și în acest caz se verifică faptul că strategiile cu funcțiile α_S și π_{RS} au o putere de selecție mai mică în comparație cu strategiile cu funcțiile α_C și α_{PRN} .

În tabela A.11 sînt date valorile funcțiilor pentru o colecție de înregistrări de lungime constantă, descriptorii avînd valorile 1, 2, 3.

Tabela A. 11

q	1 0 1 0	α_C	α_{PRN}	α_{MM}	α_S	α_{SM}	π_{RS}
x_1	1 2 3 0	$4/\sqrt{28}$	4/22	2/6	1	4/6	1/3
x_2	3 2 1 0	$4/\sqrt{28}$	4/22	2/6	1	4/6	1/3
x_3	2 1 3 0	$5/\sqrt{28}$	5/21	2/6	1	5/6	5/24
x_4	2 3 1 0	$3/\sqrt{28}$	3/23	2/6	1	3/6	3/8
x_5	0 1 2 3	$2/\sqrt{28}$	2/24	1/6	1/2	2/6	1/8
x_6	0 3 2 1	$2/\sqrt{28}$	2/24	1/6	1/2	2/6	1/8
x_7	0 2 1 3	$1/\sqrt{28}$	1/25	1/6	1/2	1/6	1/4
x_8	0 2 3 1	$3/\sqrt{28}$	3/23	1/6	1/2	3/6	1/12
x_9	1 2 0 3	$1/\sqrt{28}$	1/25	1/6	1/2	1/6	1/4
x_{10}	3 2 0 1	$3/\sqrt{28}$	3/23	1/6	1/2	3/6	1/12
x_{11}	2 1 0 3	$2/\sqrt{28}$	2/24	1/6	1/2	2/6	1/8
x_{12}	2 3 0 1	$2/\sqrt{28}$	2/24	1/6	1/2	2/6	1/8
x_{13}	1 0 2 3	$3/\sqrt{28}$	3/23	2/6	1	3/6	3/8
x_{14}	3 0 2 1	$5/\sqrt{28}$	5/21	2/6	1	5/6	5/24
x_{15}	2 0 1 3	$3/\sqrt{28}$	3/23	2/6	1	3/6	3/8
x_{16}	2 0 3 1	$5/\sqrt{28}$	5/21	2/6	1	5/6	5/24
x_{17}	3 0 1 2	$4/\sqrt{28}$	4/22	2/6	1	4/6	1/3
x_{18}	1 0 3 2	$4/\sqrt{28}$	4/22	2/6	1	4/6	1/3

În tabela A.12 sînt date răspunsurile sistemului pentru strategiile din tabela A.11.

Tabela A. 12

Funcția	Răspunsul sistemului
π_{RS}	$x_3 \ x_{14} \ x_{16}$ $x_1 \ x_2 \ x_{17} \ x_{18}$ $x_4 \ x_{13} \ x_{15}$ $x_7 \ x_9$ $x_5 \ x_6 \ x_{11} \ x_{12}$ $x_8 \ x_{10}$
α_C α_{PRN} α_{SM}	$x_3 \ x_{14} \ x_{16}$ $x_1 \ x_2 \ x_{17} \ x_{18}$ $x_4 \ x_8 \ x_{10} \ x_{13} \ x_{15}$ $x_5 \ x_6 \ x_{11} \ x_{12}$ $x_7 \ x_9$
α_S α_{MM}	$x_1 \ x_2 \ x_3 \ x_4 \ x_{13} \ x_{14} \ x_{15} \ x_{16} \ x_{17} \ x_{18}$ $x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12}$

Se observă că în acest caz strategia cu funcția π_{RS} este cea mai eficientă. Strategiile cu funcțiile α_C α_{PRN} α_{SM} sînt echivalente cu o strategie cu funcția liniară

$$\sum_{k=1}^n d_k(x) d_k(q),$$

care reprezintă un hiperplan.

ANEXA 2

METODE DE REZOLVARE A ECUAȚIEI FUNDAMENTALE DIN ANALIZA CLASELOR LATENTE

Fie ecuația

$$p_i = \sum_{i=1}^m p(x/i) p_i^i.$$

În cele ce urmează se dau două metode de rezolvare.
Cazul $m = (n+1)/2$
Considerăm matricele

$$A = \begin{bmatrix} p_0^1 & p_1^1 & p_2^1 & \cdot & \cdot & \cdot & p_n^1 \\ p_0^2 & p_1^2 & p_2^2 & \cdot & \cdot & \cdot & p_n^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ p_0^m & p_1^m & p_2^m & \cdot & \cdot & \cdot & p_n^m \end{bmatrix},$$

unde prin convenție $p_0^i = 1, i = 1, \dots, m,$

$$B = \begin{bmatrix} 1 & p_1^1 & p_2^1 & \cdot & \cdot & \cdot & p_{m-1}^1 \\ 1 & p_1^2 & p_2^2 & \cdot & \cdot & \cdot & p_{m-1}^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & p_1^m & p_2^m & \cdot & \cdot & \cdot & p_{m-1}^m \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & p_m^1 & p_{m+1}^1 & \cdot & \cdot & \cdot & p_{2m-2}^1 \\ 1 & p_m^2 & p_{m+1}^2 & \cdot & \cdot & \cdot & p_{2m-2}^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & p_m^m & p_{m+1}^m & \cdot & \cdot & \cdot & p_{2m-2}^m \end{bmatrix},$$

$$D = \begin{bmatrix} p(x/1) & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & p(x/2) & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & p(x/m) \end{bmatrix}.$$

$$H = \begin{bmatrix} p_n^1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & p_n^2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & p_n^m \end{bmatrix}.$$

Fie P matricea formată cu elementele $p_{ij/n}$, $i = 0, 1, 2, \dots, m-1$, $j=0, m, m+1, \dots, 2m-2$,

$$P = \begin{bmatrix} p_{00n} & p_{0mn} & p_{0,m+1,n} & \cdot & \cdot & \cdot & \cdot & p_{0,2m-2,n} \\ p_{10n} & p_{1mn} & p_{1,m+1,n} & \cdot & \cdot & \cdot & \cdot & p_{1,m-2,n} \\ p_{20n} & p_{2mn} & p_{2,m+1,n} & \cdot & \cdot & \cdot & \cdot & p_{2,2m-2,n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ p_{m-1,0,n} & p_{m-1,m,n} & p_{m-1,m+1,n} & \cdot & \cdot & \cdot & \cdot & p_{m-1,2m-2,n} \end{bmatrix}.$$

Fie P^* o matrice definită ca P unde $p_{ij/n}$ este înlocuit cu p_{ij}

$$P^* = \begin{bmatrix} p_{00} & p_{0m} & p_{0,m+1} & \cdot & \cdot & \cdot & \cdot & p_{0,2m-2} \\ p_{10} & p_{1m} & p_{1,m+1} & \cdot & \cdot & \cdot & \cdot & p_{1,2m-2} \\ p_{20} & p_{2m} & p_{2,m+1} & \cdot & \cdot & \cdot & \cdot & p_{2,2m-2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ p_{m-1,0} & p_{m-1,m} & p_{m-1,m+1} & \cdot & \cdot & \cdot & \cdot & p_{m-1,2m-2} \end{bmatrix}$$

Cu aceste notații ecuația fundamentală devine

$$P = B'DEC,$$

$$P^* = B'DC.$$

Fie ecuația

$$|P - \lambda P^*| = 0,$$

care se poate scrie

$$|B'DEC - \lambda B'DC| = 0$$

sau

$$|B'| |D| |E - \lambda I| |C| = 0.$$

Considerind B, C, D , ca fiind matrice neregulate, atunci deoarece E este o matrice diagonală, rădăcinile $\lambda_1, \lambda_2, \dots, \lambda_m$ sînt elementele pe diagonală matricei E , adică $p_1^1, p_2^2, \dots, p_m^m$. Fiindcă $|P - \lambda^i P^*| = 0$, matricea $(P - \lambda^i P^*)$ este singulară și deci este posibil să se găsească un vector coloană u^i care nu are toate componentele nule și care satisface condiția

$$(P - \lambda^i P^*) u^i = 0,$$

adică

$$P u^i = \lambda^i P^* u^i.$$

Fie $U = (u^1, u^2, \dots, u^m)$ și Λ o matrice diagonală cu elementele λ^i pe diagonală. Atunci

$$PU = P^* U \Lambda$$

Presupunem că λ^i sînt ordonate în astfel ca $\Lambda = E$. Atunci o soluție pentru U este $U = C^{-1}$, ceea ce se verifică ușor făcînd înlocuiri

$$B^i DECC^{-1} = B^i DCC^{-1}E.$$

Cînd elementele λ^i sînt ordonate, U este unic determinat cu excepția multiplicării la dreapta cu o matrice diagonală M . Astfel orice soluție U poate fi exprimată ca $U = C^{-1}M$. Invers, fiind dată o soluție U , $C = MU^{-1}$. Fiindcă elementele primei coloane ale matricei C trebuie să fie egale cu unitatea, fiecare element diagonal al matricei M trebuie să fie reciprocul primului element al rîndului corespunzător din U^{-1} . În felul acesta vectorii proprii ai matricei P în termenii matricei P^* determină matricea C .

Considerăm vectorul linie $(v^i)^t$ care satisface condiția

$$(v^i)^t (P - \lambda^i P^*) = 0.$$

Prin transpunere

$$P^t v^i = \lambda^i P^{*t} v^i,$$

unde

$$P^t = C^t DEB \quad P^{*t} = C^t DB.$$

Ca mai sus

$$B = M, V^{-1},$$

unde $V = (v^1, v^2, \dots, v^m)$, iar M este o matrice diagonală în care fiecare element este reciprocul primului element în rîndul corespunzător al V^{-1} .

În final se obține

$$(B^t)^{-1} P C^{-1} = (B^t)^{-1} B^t D C C^{-1} = D.$$

Metoda are dezavantajul că folosește numai o parte a informației disponibile și că matricele manipulate sînt nesimetrice.

Cazul $m=n$

Considerăm matricele

$$A = \begin{bmatrix} 1 & p_1^1 & p_2^1 & \dots & p_{n-1}^1 \\ 1 & p_1^2 & p_2^2 & \dots & p_{n-1}^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & p_1^n & p_2^n & \dots & p_{n-1}^n \end{bmatrix},$$

$$D = \begin{bmatrix} p(x/1) & 0 & \dots & 0 \\ 0 & p(x/2) & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & p(x/m) \end{bmatrix},$$

$$E = \begin{bmatrix} p_n^1 & 0 & \dots & 0 \\ 0 & p_n^2 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & p_n^m \end{bmatrix},$$

Fie P matricea formată din elementele p_{ij} , $i = 0, 1, 2, \dots, n-1$, $j = 0, 1, 2, \dots, m-1$,

$$P = \begin{bmatrix} p_{00n} & p_{01n} & p_{02n} & \cdots & p_{0,n-1,n} \\ p_{10n} & p_{11n} & p_{12n} & \cdots & p_{1,n-1,n} \\ p_{20n} & p_{21n} & p_{22n} & \cdots & p_{2,n-1,n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ p_{n-1,0,n} & p_{n-1,1,n} & p_{n-1,2,n} & \cdots & p_{n-1,n-1,n} \end{bmatrix}$$

și P^* o matrice definită ca P , unde p_{ij} este înlocuit cu p_{ji} ,

$$P^* = \begin{bmatrix} p_{00} & p_{01} & p_{02} & \cdots & p_{0,n-1} \\ p_{10} & p_{11} & p_{12} & \cdots & p_{1,n-1} \\ p_{20} & p_{21} & p_{22} & \cdots & p_{2,n-1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ p_{n-1,0} & p_{n-1,1} & p_{n-1,2} & \cdots & p_{n-1,n-1} \end{bmatrix}.$$

În acest caz ecuația fundamentală poate fi scrisă în forma

$$P = A^t D E A$$

$$P^* = A^t D A.$$

Rezolvarea ecuației

$$|P - \lambda P^*| = 0$$

se face ca în cazul precedent.

Avantajul metodei constă în faptul că matricele P și P^* sînt simetrice.

Numim stratificator descriptorul n care este adăugat elementelor matricei P^* pentru a forma matricea P .

Acest stratificator este ales astfel ca $p'_{in} \neq p'_{jn}$ pentru toți (i, j) , $i \neq j$. Pot fi multe stratificări care să satisfacă condiția de mai sus, însă datorită consistenței algebrice folosirea oricăreia din ele va conduce la același rezultat, adică la aceeași mulțime de parametri $p(x/i)$ și p'_i .

ANEXA 3

TEOREMA DE SEPARARE A MULȚIMILOR FUZZY

TEOREMĂ. În spațiul X n -dimensional fie G_1 , G_2 și $G_k = G_1 \cap G_2$, mulțimi fuzzy convexe mărginite cu

$$\begin{aligned} M_1 &= \sup \varphi_1(x), \\ M_2 &= \sup \varphi_2(x), \\ M_k &= \sup \varphi_k(x), \end{aligned}$$

atunci $1 - M_k$ este cel mai mare grad de separare al mulțimilor G_1 și G_2 , ce poate fi realizat cu un hiperplan H în X .

Demonstrație.

Cazul $M_k = \min(M_1, M_2)$. Fie $M_1 < M_2$, încît $M_k = M_1$. Atunci datorită proprietății de mulțime mărginită există un hiperplan H astfel încît

$$[\forall x \in H_-] (\varphi_1(x) \leq M_k).$$

$$H = \{x \mid ax = c\}$$

$$H_+ = \{x \mid ax > c\}$$

$$H_- = \{x \mid ax < c\}$$

Astfel

$$[\forall x \in H_+] (\varphi_1(x) \leq M_k),$$

deoarece

$$[\forall x \in X] (\varphi_1(x) = M_1 = M_k).$$

Rămîne de arătat că nu există un $M_1 < M_k$ și un hiperplan H' astfel ca

$$[\forall x \in H'_-] (\varphi_1(x) \leq M_1),$$

$$[\forall x \in H'_+] (\varphi_1(x) \leq M_1).$$

Numim nucleu mulțimea $\{x \mid M_1 = M_k\}$ și presupunem că nucleul grupării G_1 este în H'_+ . Atunci afirmația

$$[\forall x \in H'_+] (\varphi_1(x) \leq M_1)$$

nu este adevărată și deci

$$[\forall x \in H'_-] (\varphi_i(x) \leq M_r),$$

$$[\forall x \in H'_+] (\varphi_j(x) \leq M_r)$$

În consecință

$$[\forall x \in H'_+] (\sup \min (\varphi_i(x), \varphi_j(x)) \leq M_r),$$

$$[\forall x \in H'_-] (\sup \min (\varphi_i(x), \varphi_j(x)) \leq M_r).$$

În această situație

$$[\forall x \in X] (\sup \min (\varphi_i(x), \varphi_j(x)) = M_r),$$

ceea ce contravine presupunerii că

$$\sup \min (\varphi_i(x), \varphi_j(x)) = M_k > M_r.$$

Cazul $M_k < \min (M_i, M_j)$. Considerăm mulțimile convexe

$$A_i = \{x \mid \varphi_i(x) > M_k\},$$

$$A_j = \{x \mid \varphi_j(x) > M_k\}.$$

Aceste mulțimi sînt nevide și disjuncte, deoarece dacă nu este așa va fi un punct y încît $\varphi_i(y) > M_k$ și $\varphi_j(y) > M_k$, și deci $\varphi_k(y) > M_k$, ceea ce contrazice presupunerea că $M_k = \sup \varphi_k(x)$.

Fiindcă A_i și A_j sînt disjuncte, prin teorema de separare a mulțimilor convexe există un hiperplan H astfel ca A_i este în H_+ și A_j în H_- .

Prin definițiile mulțimilor A_i și A_j ,

$$[\forall x \in H_+] (\varphi_i(x) \leq M_k),$$

$$[\forall x \in H_-] (\varphi_j(x) \leq M_k).$$

Astfel a fost arătat că există un hiperplan H care realizează $1 - M_k$ separarea lui G_i de G_j . Concluzia că un grad de separare mai mare nu poate fi realizat urmează din argumentul dat în primul caz.

BIBLIOGRAFIE

1. ABRAHAM C. T., *Techniques for Thesaurus Organization and Evaluation*, in *Some Problems in Information Science*, sub red. KOCHEN M., The Scarecrow Press, New York, 1965, p. 131—151.
2. ABRAHAM C. T., *Graph-Theoretic Techniques for the Organization of Linked Data*, in *Some Problems in Information Science*, sub red. KOCHEN M., The Scarecrow Press, New-York, 1965, p. 229—252.
3. AGAMALOVA M. A., AGOPIAN R. A., *Obuceamte matritsi klasifikatori informatii*, Naucino-Tehniceskaja Informacija, 3, 22—26 (1964).
4. ANDERSON R. R., *An Associativity Technique for Automatically Optimizing Retrieval Results*, Report No. 3: *Experimental Retrieval Systems Studies*, Center for the Information Sciences, Lehigh University, April 1967.
5. ANDERSON T. W., *On Estimation of Parameters in Latent Structure Analysis*, *Psychometrika*, 19, 1, 1—12 (1954).
6. AVRAMESCU A., *Evaluation of Information Retrieval Systems Efficiency*, paper presented at FID Congress Moscow, Sept., 1968.
7. AVRAMESCU A., *Acoperirea unui domeniu printr-un vocabular de descriptori*, Studii și cercetări de documentare 1, 3—22 (1969).
8. BECKER J., HAYES R. M., *Introduction to Information Storage and Retrieval: Tools, Elements, Theories*, Wiley, New York, 1963.
9. BAKER F. B., *Information Retrieval Based Upon Latent Class Analysis*, *Journal of the Association for Computing Machinery*, 9, 4, 512—521 (1962).
10. BARNES R., *Mathematico-Logical Foundations of Retrieval Theory: General Concepts and Methods*, Center for the Information Sciences, Lehigh University, Nov. 1965.
11. BAXENDALE P., *An empirical model for computer indexing*, in *Machine Indexing: Progress and Problems*, The American University, Washington, 1961.
12. BAXENDALE P., *Machine-made Index for Technical Literature — An Experiment*, *IBM Journal of Research and Development*, 2, 4, 354—361 (1958).
13. BELZER J., GOFFMAN W., *Theoretical Considerations in Information Retrieval Systems*, *Communications of the Association for Computing Machinery*, 7, 7, 439—441 (1964).

14. BOBROW D. G., *Syntactic Theory in Computer Implementations*, in *Automated Language Processing* sub red. BORKO H., Wiley, New York, 1967, p. 215—253.
15. BONNER R. E., *On Some Clustering Techniques*, IBM Journal of Research and Development, **8**, 1, 22 (1964).
16. BORKO H., *Research in Computer Based Classification Systems*, in *Classification Research: Proceedings of the Second International Study Conference*, Elsinore, Sept. 1964, p. 220—258.
17. BORKO H., *Indexing and Classification*, in *Automated Language Processing* sub red., BORKO H., Wiley New York, 1967, p. 99—126.
18. BORKO H., BERNICK M. D., *Automatic Document Classification*, Journal of the Association for Computing Machinery, **10**, 2, 151—162 (1963).
19. BORODIN, A., KERR, L., LEWIS, F., *Query Splitting in Relevance Feedback Systems*. Information Storage and Retrieval, Report ISR — 14 to National Science Foundation, Department of Computer Science, Cornell University, October 1968.
20. BRAUEN T. L., HOLT, R. C., WILCOX, T. R., *Document Indexing Based on Relevance Feedback*. Information Storage and Retrieval, Report ISR—14 to National Science Foundation. Department of Computer Science, Cornell University, October 1968.
21. CHIEN R. T., PREPARATA F. P., *Topological Structures of Information Retrieval Systems*, Report R-348, Coordinated Sciences Laboratory, University of Illinois, May 1967.
22. CHIEN R. T., PREPARATA F. P., *Search Strategy and File Organization in Computerized Information Retrieval Systems with Mass Memory*, paper presented at the FID/IFIP Conference on Mechanized [Information Storage Retrieval and Dissemination, Rome, June 1967.
23. CLEVERDON C. W., *Automation in Indexing*, Aslib Proceedings, **13**, 4 (1961).
24. CLIMENSON W. D., HARDWICK N. H., JACOBSON S. N., *Automatic Syntax Analysis in Machine Indexing and Abstracting*, American Documentation, **12**, 3, 178—183 (1961).
25. CONSTANTINESCU P., *The Classification of a Set of Elements with Respect with a Set of Properties*, Computer Journal, January 1966, p. 352—357.
26. COOPER P. W., *Hyperplanes, Hyperspheres, Hyperquadrics as Decision Boundaries*, in *Computer and Information Sciences* sub red., TOU J., WILCOX R., Spartan Books, Washington, 1964.
27. COOPER W. S., *Expected Search Length: A Single Measure of Retrieval Effectiveness Based on the Weak Ordering Action of Retrieval Systems*, American Documentation, **19**, 1, 30—40 (1968).

28. COVER . M., *Classification and Generalization Capabilities of Linear Threshold Units*, Rome Air Development Center, Technical Documentary Report RADC-TDR-64-32, February 1964.
29. CRAWFORD R. G., MELZER, H. Z., *The Use of Relevant Documents Instead of Queries in Relevance Feedback*, Information Storage and Retrieval, Report ISR - 14 to National Science Foundation, Department of Computer Science, Cornell University, October 1968.
30. CURTICE R. M., *Magnetic Tape and Disc File Organizations for Retrieval*, Report No. 1: *Experimental Retrieval Systems Studies*, Center for the Information Sciences, Lehigh University, July 1966.
31. CURTICE R. M., ROSENBERG V., *Optimizing Retrieval Results with Man-Machine Interaction*, Center for the Information Sciences, Lehigh University 1965.
32. DALE A. G., *Discussion*, in *Classification Research: Proceedings of the Second International Study Conference*, Elsinore, Sept. 1964, p. 262.
33. DAVIS, M. C., LINSKY, M. D., ZELKOWITZ, M. V., *A Relevance Feedback System Employing a Dynamically Evolving Document Space*, Information Storage and Retrieval, Report ISR - 14 to National Science Foundation, Department of Computer Science Cornell University, October, 1968.
34. DINULEANU N., *Teoria măsurii și funcții reale*, Edit. didactică și pedagogică, București, 1964.
35. DOYLE L. B., *Indexing and Abstracting by Association*, American Documentation, 13, 4, 378-390 (1962).
36. DOYLE L. B., *Semantic Road Maps for Literature Searches*, Journal of the Association for Computing Machinery, 8, 4, 553-578 (1961).
37. DOYLE L. B., *Is Automatic Classification a Reasonable Application of Statistical Analysis of Text?*, Journal of the Association for Computing Machinery, 12, 4, 473-489 (1965).
38. EDMUNSON H. P., *Mathematical Models in Linguistics and Language Processing in Automated Language Processing*, sub red., BORKO H., Wiley, New York, 1967.
39. EDMUNSON H. P., WYLLYS R. E., *Automatic Abstracting and Indexing. Survey and Recommendations*, Communications of the Association for Computing Machinery, 4, 5, 226-234 (1961).
40. FAIRTHORNE R. A., *Toward Information Retrieval*, Butterworth's, Londra, 1965.
41. FALKOFF A. D., *Algorithms for Parallel-Search Memories*, Journal of the Association for Computing Machinery, 9, 4, 488 (1962).
42. FRAZER W. D., *A proposed System for Multiple Descriptor Data Retrieval*, in *Some Problems in Information Science* sub red., KOCHEN M., The Scarecrow Press, New York, 1965, p. 187-206.

43. GARFIELD E., *Citation Indexes for Science*, Science **122**, 3159, 108—111 (1955).
44. GARFIELD E., *Science Citation Index — A New Dimension in Indexing*, Science, **144**, 3619, 649—654 (1964).
45. GHELFAND I. M., *Lecții de algebră liniară*, Edit. tehnică, București, 1953.
46. GHICA AL., *Analiza funcțională*, Edit. Academiei, București, 1967.
47. GIBSON W. A., *An Extension of Anderson's Solution for the Latent Structure Equations*, Psychometrika, **20**, 1, 60—73 (1955).
48. GIULIANO V. E., JONES P. E., *Study and Test for a Methodology for Laboratory Evaluation of Message Retrieval Systems*, Arthur D. Little Inc., Report ESD-TR-66—405 August 1966.
49. GIULIANO V. E., JONES P. E., *Linear Associative Information Retrieval*, in *Vistas in Information Handling*, sub red., HOWERTON P., WEEKS D., Spartan Books, Washington, 1963, p. 30—46.
50. GOFFMAN W., *A Searching Procedure for Information Retrieval*, Information Storage and Retrieval, **2**, 73—78 (1964).
51. GOFFMAN W., *On Relevance as a measure*, Information Storage and Retrieval, **2**, 201—203 (1964).
52. GOFFMAN W., *On the Logic of Information Retrieval*, Information Storage and Retrieval, **2**, 217—220 (1964).
53. GOFFMAN W., NEWILL V. A., *A Methodology for Test and Evaluation of Information Retrieval Systems*, Information Storage and Retrieval, **3**, 19—25 (1966).
54. GOODMAN N., *Axiomatic Measurement of Simplicity*, The Journal of Philosophy, **12**, 24, 702—722 (1955).
55. HALMOS P., *Measure Theory*, Van Nostrand, Princeton, 1956.
56. HARRAND Y., *Trailement des files et des listes*, Dunod, Paris, 1967.
57. HARARY F., NORMAN R. Z., CARTWRIGHT D., *Structural Models. An Introduction to the Theory of Directed Graphs*, Wiley, New York, 1965.
58. HAYES R. M., *A Theory for File Organization*, in *On-line Computing*, sub red. KARPLUS W., McGraw-Hill, New York, 1965.
59. HEWITT E., STROMBERG K., *Real and Abstract Analysis*, Springer, Berlin, 1965.
60. HIGHLEYMAN W. H., *Linear Decision Functions with Application to Pattern Recognition*, Proceedings of the IRE, **50**, 6, 1501—1514 (1962).
61. HILL D. R., *A vector Clustering Technique*, paper presented at the FID/IFIP Conference on Mechanized Information Storage Retrieval and Dissemination, Rome, June 1967.
62. HILLMAN D., *Two Models for Retrieval System Design*, American Documentation, **15**, 3, 217—225 (1964).
63. HILLMAN D., *Characterization and Connectivity*, Report No. 1 : *Document Retrieval Theory, Relevance and the Methodology of Evaluation*, Center for the Information Sciences, Lehigh University, 1966.

64. HILLMAN D., *Grammars and Text Analysis*, Report No. 1 : *Computational, Phonological and Morphological Linguistics and Retrieval Studies*, Center for the Information Sciences, Lehigh University, August 1965.
65. HILLMAN D., REED D. M., *Microcategorization for Text Processing*, Report No. 3 : *Document Retrieval Theory, Relevance and the Methodology of Evaluation*, Center for the Information Sciences, Lehigh University, 1966.
66. HOWARD R. N., *Classifying a Population into Homogenous Groups*, paper presented at Conference of Operational Research Society, Cambridge, 1964.
67. IDE, E., *New Experiments in Relevance Feedback*. Information Storage and Retrieval, Report ISR - 14 to National Science Foundation, Department of Computer Science, Cornell University, October 1968.
68. IVERSON K. E., *A Programming Language*, Wiley, New York, 1962.
69. JONES P. E., CURTICE R. M., *A Framework for Comparing Term Association Measures*, American Documentation, July 1967, p. 153.
70. KASARDA A. J., *A Syntactically Oriented Natural Language Document Retrieval System with a Browsability Feature*, Report No. 3 : *Experimental Retrieval Systems Studies*, Center for the Information Sciences, Lehigh University, April 1967.
71. KEEN E. M., *Semi-Automatic User-Controlled Search Strategy*, paper presented at the Fourth Annual Colloquium on Information Retrieval, Philadelphia, May 1967.
72. KENT A., *Textbook on Mechanized Information Retrieval*, Wiley, New York, 1962.
73. KENT A., TAULBEE O., (ed), *Electronic Information Handling*, Spartan Books, Washington, 1965.
74. KESSLER M. M., *An Experimental Study of Bibliographic Coupling Between Technical Papers*, MIT-Report-1, November 1961.
75. KRAIZMER, L., BORODAEV A., GUTENMACHER L., KUZMIN B., SMULIANSKI I., *Asotiativnye Zapominaiushchie Ustroistva*, Energhia, Leningrad, 1967.
76. KRAVETS L. G., MOSCOVICI V. A., SENDEROV V. A., *Automatic Indexing of Patent Information*, paper presented at ICIREPAT Meeting, The Hague, October 1966.
77. LANDAUER W. I., *The Tree as a Stratagem for Automatic Information Handling*, A Dissertation in Electrical Engineering presented to the University of Pennsylvania in partial fulfilment of the requirements for the degree of Doctor of Philosophy, 1962.
78. LAZARSFELD P. F., STOUFFER S. A., *Measurement and Prediction*, Princeton University Press, 1950.
79. LEFKOVITZ D., *Automatic Stratification of Descriptors*, A Dissertation in Electrical Engineering, University of Pennsylvania, The Moor School of Electrical Engineering, 1963.
80. LE SCHACK R., *A Note on Measures of Similarity*, Report No. ISR-7, Information Storage and Retrieval, The Computation Laboratory of Harvard University, June 1964.

81. LE SCHACK R., *The Determination of Clusters by Matrix Analysis*, Report No. ISR-7, Information Storage and Retrieval, The Computation Laboratory of Harvard University, June 1964.
82. LESK M. E., *Procedures for Statistical Processing and Request Alteration*, Report No. ISR-7, Information Storage and Retrieval, The Computation Laboratory of Harvard University, June 1964.
83. LESK M. E., *Word-Word Associations in Document Retrieval Systems*, Report No. IRS-13, Information Storage and Retrieval, Department of Computer Science, Cornell University, January 1968.
84. LESK M. E., SALTON G., *Design Criteria for Automatic Information Systems*, Report No. IRS-11, Information Storage and Retrieval, Department of Computer Science, Cornell University, June 1966.
85. LESSER V. R., *A Modified Two-Level Search Algorithm Using Request Clustering*, Report No. IRS-11, Information Storage and Retrieval, Department of Computer Science, Cornell University, June 1966.
86. LEVERY F., *Un experience d'indexage automatique*, Congressorasegna internazionale sulla documentazione l'informazione scientifico-tecnica, Roma, 1961.
87. LEWIS P. A., BAXENDALE P. B., BENETT J. L., *Statistical Discrimination of the Synonymy/Antonymy Relationship Between Words*, Journal of the Association for Computing Machinery, **14**, 1, 20 (1967).
88. LOWE T. C., *Design Principles for an On-line Information Retrieval System*, Technical Report, The Moore School of Electrical Engineering, University of Pennsylvania, December 1966.
89. LUHN H. P., *A Statistical Approach to Mechanical Encoding and Searching of Library Information*, IBM Journal of Research and Development, **1**, 4, 309-317 (1957).
90. LUHN H. P., *Keyword-Context Index for Technical Literature (KWIC Index)*, American Documentation, **11**, 228-295 (1960).
91. LUSTIG G., *A New Class of Association Factors*, paper presented at the FID/IFIP Conference on Mechanized Information Storage Retrieval and Dissemination, Rome, June 1967.
92. MANDANSKY A., *Determinantal Methods in Latent Class Analysis*, Psychometrika, **25**, 2, 183-197 (1960).
93. MARCUS S., NICOLAU ED., STATI S., *Introducere în lingvistica matematică*, Edit. științifică, București, 1966.
94. MARCUS S., *Gramatlei și automate finite*, Edit. Academiei, București, 1964.
95. MARON M. E., *Mechanized Documentation: The Logic Behind a Probabilistic Interpretation*, paper presented at the Symposium on Statistical Association Methods for Mechanical Documentation, sponsored by the National Bureau of Standards, held at the Smithsonian Institute, March 17-19, 1964.

96. MARON M. E., *The Logic of Interrogating a Digital Computer*, paper presented at the 1964 Linguistic Institute of the Linguistic Society of America, held at the University of Indiana.
97. MARON M. E., *Automatic Indexing; an Experimental Inquiry*, Journal of the Association for Computing Machinery, 8, 3, 404—417 (1961).
98. MARON M. E., KUHN J. L., *On Relevance, Probabilistic Indexing and Information Retrieval*, Journal of the Association for Computing Machinery, 7, 3, 216—244 (1960).
99. MEADOW C. T., *The Analysis of Information Systems*, Wiley, New York, 1967.
100. MEETHAM A., *Graph Separability and Word Grouping*, in *Proceedings of 21-st National Conference of the Association for Computing Machinery*, Academic Press, New York, 1967, p. 513—514.
101. MIHAILOV A. I., CERNÎI A., GHILIAREVSKI R., *Osnovi Naucinoi Informații*, Izdatelstvo Nauka, Moscova, 1965.
102. MOOERS C., *A Mathematical Theory of Language Symbols in Retrieval*, in *Proceedings of the International Conference on Scientific Information*, Washington, 1959, vol. 2, p. 1327—1364.
103. NEEDHAM R. M., *The Theory of Clumps II*, Report ML 139, The Cambridge Language Research Unit, March 1961.
104. NEEDHAM R. M., JACKSON D. M., SPARCK JONES K., *Notes on the Extension of Clump-Finding Techniques*, Report ML 189, The Cambridge Language Research Unit, May 1966.
105. NEEDHAM R. M., SPARK JONES K., *Keywords and Clumps*, Journal of Documentation, 20, 1 (1964).
106. NEGOIȚĂ C. V., *Asupra automatizării documentării*, Comunicare la sesiunea tehnico-științifică OSI, mai 1963.
107. NEGOIȚĂ C. V., *Studiu privind mecanizarea documentării din literatura de brevete*, Oficiu de stat pentru invenții, decembrie 1966.
108. NEGOIȚĂ C. V., *Utilizarea calculatorului numeric în stabilirea diagnosticului*, Automatica și electronica, 9, 1, 35—39 (1966).
109. NEGOIȚĂ C. V., *Utilizarea calculatorului electronic numeric pentru diagnostic*, Viața medicală, XIII, 17, 1203—1206 (1966).
110. NEGOIȚĂ C. V., *Tendențe privind mecanizarea documentării din brevete*, Invenții și inovații, 1, 9, 343—345 (1966).
111. NEGOIȚĂ C. V., *Asupra strategiei de căutare de tip probabilistic pentru sistemele de regăsire a informațiilor*, Studii și cercetări de calcul economic și cibernetică economică, 6, 47—52 (1967).
112. NEGOIȚĂ C. V., *Studiu privind sistemele de informare bazate pe calculator*, Academia Republicii Socialiste România, Centrul de documentare științifică, 1967.
113. NEGOIȚĂ C. V., *Calculatoarele electronice și aplicațiile lor în economie*, Progresele științei, 3, 11, (1967).

114. NEGOIȚĂ C. V., *Strategy of Probabilistic Type for Information Retrieval Systems*, Economic Computation and Economic Cybernetics Studies and Research, 2, (1968).
115. NEGOIȚĂ C. V., *Considerații teoretice privind sistemele de regăsire a informațiilor, bazate pe calculator electronic*, Studii și cercetări de documentare și bibliologie, 1 (1968).
116. NEGOIȚĂ C. V., *Metode de clasificare automată în sistemele de regăsire a informațiilor*, Studii și cercetări de documentare și bibliologie, 3, (1968).
117. NEGOIȚĂ C. V., *Asupra utilizării factorilor de asociere în sistemele automate de regăsire a informațiilor*, Automatica și electronica, 12, 6, (1968).
118. NEGOIȚĂ C. V., *On the Utilization of Distance Functions as Linear Discriminant Functions in Information Retrieval Systems*, Report, August 1968.
119. NEGOIȚĂ C. V. *Search Strategies in Automatic Information Systems*, paper presented at the Sixth International Congress on Cybernetics, Namur, September 1970.
120. NEGOIȚĂ C. V., *Metode de clasificare automată în sistemele de regăsire a informațiilor*, Automatică, metrologie, calculatoare, 12 (1969).
121. NEGOIȚĂ C. V., *O funcție pătratică de selecție pentru sistemele de regăsire a informațiilor*, Studii și cercetări de documentare 2 (1969).
122. NEGOIȚĂ C. V., *Asupra strategiilor de selecție în sistemele de regăsire a informațiilor*, Studii și cercetări de documentare 4 (1969).
123. NEGOIȚĂ C. V., *On the Application of the Fuzzy Sets Separation Theorem for Automatic Classification in Information Retrieval Systems*, Report, August 1969.
124. NEGOIȚĂ C. V., *Studiul sistemelor de selecție a informațiilor*, teză de doctorat, Institutul politehnic București, 1969.
125. NEGOIȚĂ C. V., *Asupra aplicării teoremei de separare a mulțimilor fuzzy în clasificarea automată*, Studii și cercetări de documentare, 1 (1970).
126. NEGOIȚĂ C. V., *Asupra procesului de indexare automată în sistemele de informare*, Automatica și electronica, 3, (1970).
127. NICOLAU ED., POPOVICI AL., *Introducere în cibernetica sistemelor discrete*, Edit. tehnică, București, 1966.
128. NICOLESCU MIRON, *Funcții reale și elemente de topologie*, Edit. didactică și pedagogică, București, 1968.
129. NIELSON N. R., *The Simulation of Time Sharing Systems*, Communication of the ACM, 10, 7 (1967).
130. NILLSON N. J., *Learning Machines*, McGraw-Hill, New York, 1965.
131. O'CONNOR J., *Mechanized Indexing Methods and Their Testing*, Journal of the ACM, 11, 4 (1964).
132. O'CONNOR J., *Automatic Subject Recognition in Scientific Papers*, Journal of the ACM, 12, 4 (1965).

133. OLMER J., *A Flexible Direct File Approach to Information Retrieval on an IBM 1401*, in *Proceedings of the 1963 Fall Joint Computer Conference*, Spartan Books, Washington, 1963.
134. ONICESCU O., *Calculul probabilităților*, Edit. didactică și pedagogică, București, 1963.
135. ONICESCU O., MIHOC G., IONESCU TULCEA C. T., *Calculul probabilităților și aplicații*, Edit. Academiei, București, 1956.
136. PARKER-RHODES A. F., *Contribution to the Theory of Clumps*, Report ML 138, The Cambridge Language Research Unit, March 1961.
137. PARKER-RHODES A. F., NEEDHAM R. M., *The Theory of Clumps: A New Concept of Classification and Selection*, Report ML 126, The Cambridge Language Research Unit, February 1960.
138. PREPARATA F. P., CHIEN R. T., *On Clustering Techniques of Citation Graphs*, Report R-349, Coordinated Sciences Laboratory, University of Illinois, May 1967.
139. PRICE N., SCHIMINOVICH S., *A Clustering Experiment: First Step Towards a Computer-Generated Classification Scheme*, *Information Storage and Retrieval*, 4, 3, 271—280 (1968).
140. RAZAR M., SHAPIRO G., *Hierarchy Set-up and Hierarchy and Concept-Concept Expansion Procedures*, Report No. ISR-9, Information Storage and Retrieval, The Computation Laboratory of Harvard University, August 1965.
141. REED D. M., *Phrase Indexing*, Report No. 3: *Experimental Retrieval Systems Studies*, Center for the Information Sciences, Lehigh University, April 1967.
142. REICH D. L., *Associative Memories and Information Retrieval*, in *Some Problems in Information Science*, sub red. KOCHEN M., The Scarecrow Press, New York, 1965, p. 217—225.
143. REISNER P., *A Note on Minimizing Search and Storage in a Thesaurus Network by Structural Reorganization of the Net*, in *Some Problems in Information Science*, sub red. KOCHEN M., The Scarecrow Press, New York, 1965, p. 265—271.
144. REITSMA K., SAGALYN J., *Correlation Measures*, Report No. IRS-13, Information Storage and Retrieval, Department of Computer Science, Cornell University, January 1968.
145. RIDDLE W., HORWITZ T., DIETZ R., *Relevance Feedback in an Information Retrieval System*, Report No. IRS-11, Information Storage and Retrieval, Department of Computer Science, Cornell University, June 1966.
146. ROCCHIO J., *Performance Indices for Document Retrieval Systems*, Report No. ISR-8, Information Storage and Retrieval, The Computation Laboratory of Harvard University, December 1964.

147. Rocchio I. J., *Document Retrieval Systems — Optimization and Evaluation*. Report ISR-10 to National Science Foundation, Harvard Computation Laboratory, March 1966.
148. ROLLING L., PIETTE J., *Interaction of Economics and Automation in a Large-size Retrieval System*, paper presented at the FID/IFIP Conference on Mechanized Information Storage Retrieval and Dissemination, Rome, June 1967.
149. ROMEIRO G. F., CAVARA L., *Assesment Studies of Documentation Systems*, Information Storage and Retrieval, 4, 3, 309—325 (1968).
150. RUBENSTEIN H., GOODENOUGH J. B., *Contextual Correlates of Synonymy*, Communications of the Association for Computing Machinery, 8, 10 (1965).
151. RUBINOFF M., *Toward a National Information System*, Second Annual National Colloquium on Information Retrieval, Spartan Books, Washington, 1965.
152. RUS T., *Folosirea teoriei arborilor la rezolvarea unor probleme nearitmetice cu ajutorul mașinilor aritmetice de calcul*, disertație pentru obținerea titlului de doctor în științe fizico-matematice, Academia Republicii Socialiste România, Filiala Cluj, Institutul de calcul, 1965.
153. SALTON G., *The Evaluation of Computer-Based Information Retrieval Systems*, in *Proceedings 1965 International FID Congress*, Spartan Books, Washington, 1966.
154. SALTON G., *Associative Document Retrieval Techniques Using Bibliographic Information*, Journal of the Association for Computing Machinery, 10, 4, 440 (1963).
155. SALTON G., *The Evaluation of Automatic Retrieval Procedures*, Report No. ISR-8, Information Storage and Retrieval, The Computation Laboratory of Harvard University, December 1964.
156. SALTON G., *Automatic Information Processing in Western Europe*, Science, 144, 3619, 626—632 (1964).
157. SALTON G., *Search and Retrieval Experiments in Real-Time Information Retrieval*. Information Storage and Retrieval, Report ISR-14 to National Science Foundation, Department of Computer Science, Cornell University, October 1968.
158. SALTON G., *Progress in Automatic Information Retrieval*, in IEEE Spectrum, August 1965, p. 90—103.
159. SALTON G., *Data Manipulation and Programming Problems in Automatic Information Retrieval*, Communications of the Association for Computing Machinery, 9, 3, 204—210 (1966).
160. SALTON G., *Information Dissemination and Automatic Information Systems*, Proceedings of the IEEE, 54, 12, 1633 (1966).
161. SALTON G., *Search Strategy and the Optimization of Retrieval Effectiveness*, paper presented at the FID/IFIP Conference on Mechanized Information Storage Retrieval and Dissemination, Rome, June 1967.

162. SALTON G., *Automatic Information Organization and Retrieval*, McGraw-Hill, New York, 1968.
163. SALTON G., LESK M. E., *Computer Evaluation of Indexing and Text Processing*, Report No IRS-12, Information Storage and Retrieval, Department, of Computer Science, Cornell University, June 1967.
164. SALTON G., LESK M. E., *Computer Evaluation of Indexing and Text Processing*, Journal of the Association for Computing Machinery, 15, 1, 8-36 (1968).
165. SALTON G., SUSSENGUTH H., *Some Flexible Information Retrieval Systems Using Structure Matching Procedures*, in *Proceedings of the AFIPS Spring Joint Computer Conference*, Spartan Books, Washington, 1964.
166. SANDERS J., *Document Association and Classification Based on L-Languages*, Journal of the Association for Computing Machinery, 12, 2, 249-253 (1965).
167. SCHULTZ L., *Language and Computer*, in *Automated Language Processing*, sub red. BORKO H., Wiley, New York, 1967, p. 11-32.
168. SHARP J., *Some Fundamentals of Information Retrieval*, Deutch, Londra, 1965.
169. SIMMONS G., *Introduction to Topology and Modern Analysis*, McGraw-Hill, New York, 1963.
170. SIMMONS R. F., *Answering English Questions by Computer*, in *Automated Language Processing*, sub red. BORKO H., Wiley, New York, 1967, p. 252-291.
171. SIMMONS R. F., McCONLOGUE K. L., *Maximum-depth Indexing for Computer Retrieval of English Language Data*, Report SP-775, System Development Corp., Santa Monica, Calif., April 1962.
172. SINETT J. D., *An Evaluation of Links and Roles in Information Retrieval*, Technical Documentary Report No. ML TDR 64-152, AF Materials Laboratory Research and Technical Division, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio, July 1964.
173. SOERGEL D., *Mathematical Analysis of Documentation Systems. An Attempt to a Theory of Classification and Search Request Formulation*, Information Storage and Retrieval, 3, 3, 129-173 (1967).
174. SOERGEL D., *Some Remarks on Information Languages Their Analysis and Comparison*, Information Storage and Retrieval, 3, 4, 219-291 (1967).
175. SPARCK JONES K., *Automatic Term Classification and Information Retrieval*, paper presented at the IFIP Congress, Edinburgh, August 1968.
176. SPARCK JONES K., JACKSON D., *The Use of the Theory of Clumps for Information Retrieval*, Report ML 190, The Cambridge Language Research Unit, June 1966.
177. SPARCK JONES K., JACKSON D., *Some Experiment in the Use of Automatically-Obtained Term Clusters for Retrieval*, paper presented at the FID/IFIP Conference on Mechanized Information Storage Retrieval and Dissemination, Rome, June 1967.

178. SPĂTARU AL., *Teoria transmisiunii informației*, Ed. tehnică, București, 1965.
179. STANFEL L. E., *On a Quantitative Approach to Improving Document Retrieval Performance*, *Information Storage and Retrieval*, 4, 3, 281—286 (1968).
180. STEINBUCH K., PISKE A. W., *Learning matrices and Their Applications*, *Transactions IEEE on Electronic Computers*, Ec-12, 5, 846—862 (1963).
181. STEVENS M. E., *Automatic Indexing: A State of the Art Report*, NBS Monograph 91, National Bureau of Standards, Washington, March 1965.
182. STEVENS M. E., GIULIANO V. E., HEILPRIN L. B. (sub red.), *Statistical Association Methods for Mechanized Documentation*, National Bureau of Standards, Publication 269, 1965.
183. STILES H. E., *The Association Factor in Information Retrieval*, *Journal of the Association for Computing Machinery*, 8, 2, 271—279 (1961).
184. VASWANI P. K., *A Technique for Cluster Emphasis and its Application to Automatic Indexing*, paper presented at the IFIP Congress, Edinburgh, August, 1968.
185. VERHOEFF J., GOFFMAN W., BELZER J., *Inefficiency of the Use of Boolean Functions for Information Retrieval Systems*, *Communications of the Association for Computing Machinery*, 4, 12, 557—559 (1961).
186. WARHEIT I. A., *File Organization for Information Retrieval*, paper presented at the FID/IFIP Conference on Mechanized Information Storage Retrieval and Dissemination, Rome, June 1967.
187. WILLIAM R., *Digital Storage Systems*, Spon, Londra, 1964.
188. WILLIAMS J. H., *Computer [Classification of Documents]*, paper presented at the FID/IFIP Conference on Mechanized Information Storage Retrieval and Dissemination, Rome, June 1967.
189. WINTERS W. K., *A Modified Method of Latent Class Analysis for File Organization in Information Retrieval*, *Journal of the Association for Computing Machinery*, 12, 3, 356—363 (1965).
190. ZADEH L. A., *Fuzzy Sets*, *Information and Control*, 8, 3, 338—353 (1965).

INFORMATION STORAGE AND RETRIEVAL SYSTEMS

BY

C. V. NEGOITA

This book deals with the computer processing of large information files, and the design of automatic information systems, with special emphasis on search strategies and automatic classification methods.

Chapter 1 presents the most significant results of the automatic indexing designed to replace a given document or search request by a vector of content identifiers (descriptors).

Chapter 2 deals with a new abstract model of the retrieval process deriving a new theory of information retrieval based on real-valued functions theory and fuzzy sets theory.

A search system is defined as a system (X, D, V, S, γ) formed with four nonempty sets and a function defined on these sets. X is the set of document vectors, D is the set of descriptors, V is the set of descriptor weights, S is the set of descriptor connections and γ is the search function of the system. This function establishes a mapping of the set X into the real line

$$\gamma: X \rightarrow R.$$

Thus, the answer of the search system is defined as the graph of the function

$$\{(x, \gamma(x)) \mid x \in X\}.$$

Denoting $P(D)$ the set of all subsets of D and μ^* the outer measure defined on $P(D)$ since $X \subset P(D)$ then if $x \in X \rightarrow \mu^*(x) = \gamma(x)$ the function γ is the restriction of μ^* .

Similarly it is demonstrated that the function γ is a step function on $P(X)$.

Considering a search query as a subset $q \subseteq D$ the search strategy is defined as the couple $\sigma = (q, \gamma)$. In this way for every strategy exists a search function

$$\gamma_\sigma(\omega) = \gamma(\omega, q).$$

In this work the search strategies are classified following the search function type: search strategies with nearness functions, search strategies with additive functions and search strategies with density functions.

Considering the commanded answer

$$X_c = \{\omega \mid \gamma_\sigma(\omega) > c_c\}$$

the search strategy effects a dichotomy i.e. divides the set X in two subsets X_c and $\complement X_c$ by the hypersurfaces $\gamma_\sigma(\omega) = c_c$. One measure of the effectiveness of a search strategy is the total number of dichotomies that its function could effect. The number of dichotomies that can be implemented by hypersurfaces depends only on the number of points $\omega \in X$ and the number of parameters of the functions defining the hypersurfaces. Thus, the quadric functions are much more powerful decision functions than are the linear functions i.e. the search strategy with density functions is the most powerful strategy.

The retrieval system illustrated in chapter 2 is based completely on individual descriptors. No relations between descriptors are assumed and the indexing operation must be completely consistent in order to obtain effective retrieval action. Chapter 3 takes into consideration for retrieval purposes the statistical association between descriptors and documents and demonstrates that in this case the strategy is automatically optimized.

Chapter 4 covers some procedures for file organization used to store the information items. Direct, inverted and multilist file organization are examined, together with the associative memory organization.

The complexity of an information retrieval task depends largely on the physical location of the documents in the file. Thus it appears quite desirable to locate physically close in the memory structures documents that are likely to be wanted together.

Chapter 5 is devoted to an examination of automatic classification methods, with special emphasis on matrix eigenvalue ana-

lysis, clump theory, discriminant functions analysis and graph theory.

Chapter 5 introduces a new clustering technique using the concept of fuzzy set. A cluster is defined as a subset $G \subset X$ characterized by a membership function $\varphi_G: X \rightarrow R$ which associates with each $x \in X$ a real number in the interval $[0, 1]$. With the values $\varphi_G(x)$ one can form the matrices $M_1 = (\varphi_i(x_k))$ and $M_2 = (\varphi_{ij}(x_k))$ where $\varphi_{ij}(x) = \min(\varphi_i(x), \varphi_j(x))$. Following the separation theorem of fuzzy sets a cluster is defined by

$$[\forall x \in G] (\varphi_k(x) < \min \max \varphi_{ij}(x)).$$

Finally chapter 5 presents the new experiments in relevance feedback.

The book serves both as a monograph for the professional practitioner versed in general computer utilization, and as a book for students enrolled in applied mathematics.

6272

Lei 7,25

EDITURA ACADEMIEI REPUBLICII SOCIALISTE ROMANIA